ALCBEAM

Neutral beam formation and propagation code

USER MANUAL

Igor Bespamyatnov:bespam@physics.utexas.edu

Note: given manual was written for ALCBEAM v. 4.6. All later updates are included in UPDATES section.

Contents

1. Updates since ALCBEAM v. 4.6.

No new updates yet.

2. Introduction

ALCBEAM is a three-dimensional neutral beam formation and propagation code. It is developed as a supplemental tool for beam-based plasma diagnostics at Alcator C-Mod tokamak. It can be also used for simulation of the neutral beams used on other tokamaks or magnetic fusion devices. The code effectively unifies the beam formation and extraction processes with beam attenuation and excitation in plasma and residual gas in the beam duct. The goal of the code is to provide reliable estimates of the local beam equilibrium parameters: such as beam energy fractions, density profiles and excitation populations. The extraction model uses the semi-empirical description to simulate the processes in the beam ion source which effect the beam formation. ALCBEAM's driver module, available from ALCBEAM widget, selects appropriate analysis time intervals for list of selected plasma shots and executes ALCBEAM base code for all of them. In such way a time dependent 3D spatial model of the beam is calculated.

The ALCBEAM code is written in IDL. The most recent version of the code is available through Alcator C-Mod PSFC workstations: /usr/local/cmod/codes/dnb/alcbeam/alcbeam.pro or through GitHub.

Starting the code:

Change to the ALCBEAM directory (for C-mod local user)

$ cd /usr/local/cmod/codes/dnb/alcbeam/

or directory where alcbeam.pro is located (for any other user):

$ cd /home/$USER/alcbeam/

Open a new IDL session:

$ idl

Compile the code:

IDL> .r alcbeam.pro

Run the ALCBEAM procedure:

IDL> alcbeam

Wait for GUI to initialize. When finished, the status log window at the bottom of the GUI will show:

:->>> Initializing …

1 : [date] Ready !!!

3. User Interface:

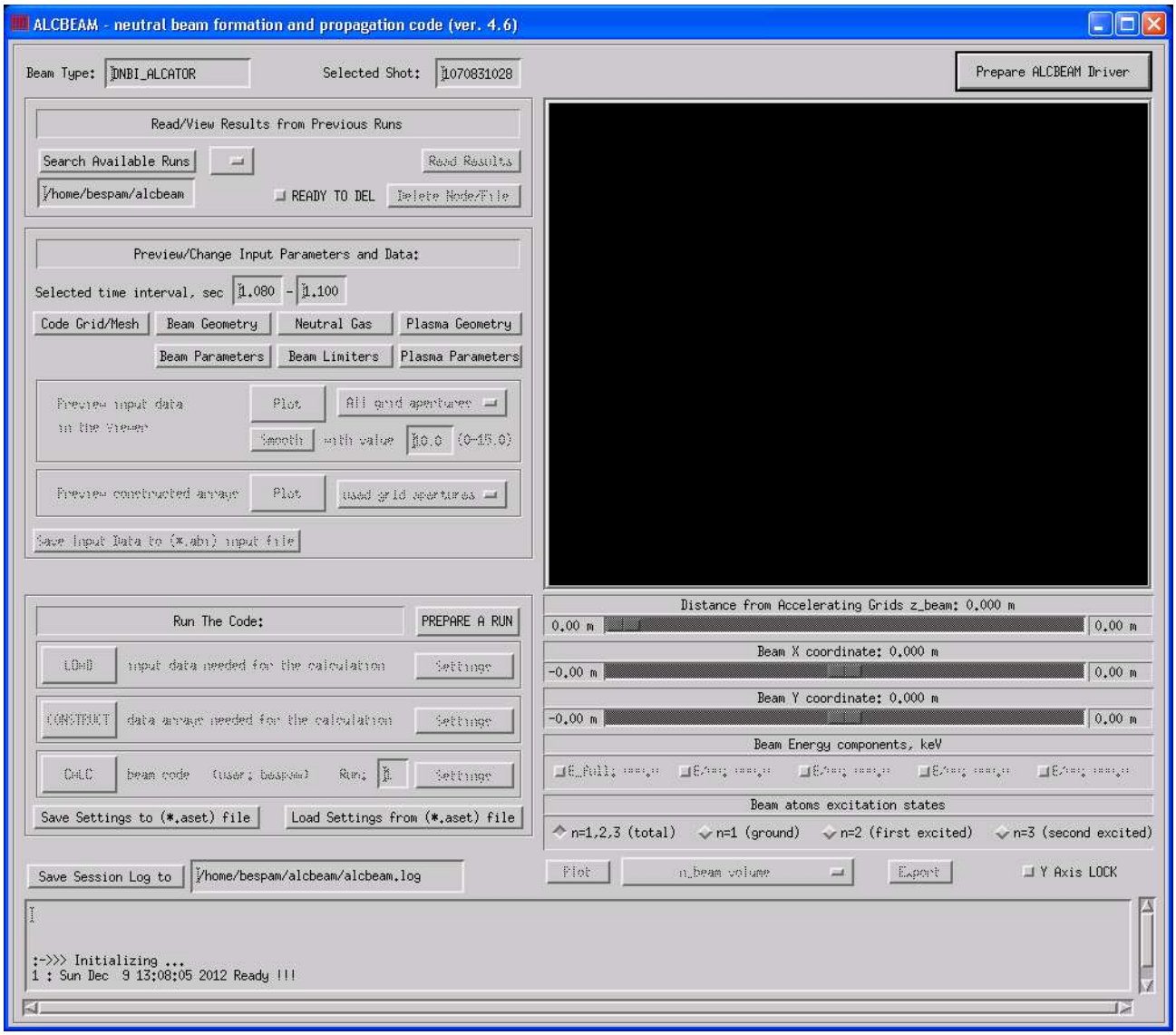


Figure 3.1.1 Main widget window

The main widget (Figure 3.1.1) is the environment in which the user can run a simulation or load results from previous simulations. The version of the code is displayed in the title of the window.

There are three directions user may precede from here. First, the user can search for available ALCBEAM simulations/runs and view results from them. A step-by-step description of this process is given in chapters 3.1-3.4. Second, a detailed description of how to initiate a new simulation is given in chapter 3.5. A third option is to launch the ALCBEAM driver. The ALCBEAM driver is used to run

multiple ALCBEAM simulations for a range of shots. For each shot, ALCBEAM driver automatically identifies a list of time intervals for which to run the simulation. Instructions of how to run ALCBEAM driver are given in chapter 3.7.

3.1 Load results from previous simulations

The user is able to load and view the results of previous simulations. For this purpose the following area of the window is used (Figure 3.1.2).

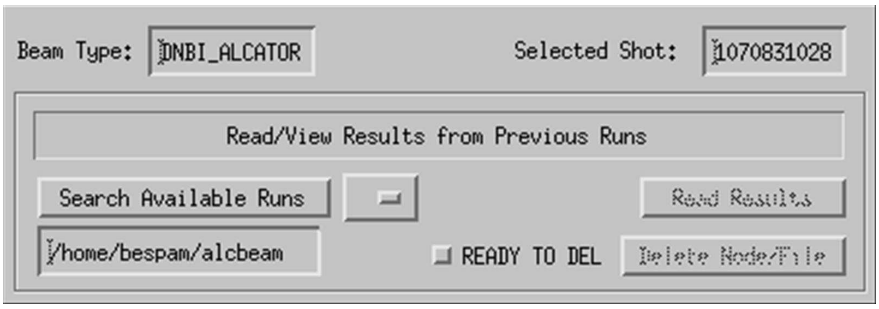


Figure 3.1.2 Load results from previous simulations

The results of the simulation can be in form of the **output_file** "*.abo" or tree in the MDSPLUS database (referred as RUNs later in the manual). Only simulation of the DNBI_ALCATOR can be saved to Alcator MDSPLUS.

The "Beam Type:" field defines the beam/system for which simulation is to be/was performed.

The "Selected Shot:" field defines for which plasma shot simulation is to be/was performed. In the case of "DNBI_ALCATOR" Beam Type, it also identifies which MDSPLUS tree to access.

The "Search Available Runs" button will look for all available simulation runs (*.abo **output_files**) saved in the target directory (displayed in the lower left corner in Figure 3.1.2). In the case of "DNBI_ALCATOR" Beam Type, it also searches for simulation RUNs saved in the MDSPLUS tree of "Selected Shot" number. The list of available runs will appear on the right side of the button (Figure 3.1.3). Runs saved in the MDSPLUS database are listed first (example of the name: BESPAM.RUN_2, which means that this is one of the RUNs performed by user BESPAM, labeled 2), followed by "*.abo" **output_files**. The format of the data saved in the MDSPLUS tree or **output_files** is identical.

A "Session Log" at the bottom of the main window displays some additional information about last action performed by the user.

After "Search Available Runs" is pressed, "Session Log" will display the number of runs found in MDSPLUS and number of files found in the selected directory. Additional information about the currently selected **run**/**output_file** is displayed (Figure 3.1.4). A more detailed explanation of this information is given later.

Figure 3.1.3 Search available results

Figure 3.1.4 Status window

After a certain **RUN**/**output_file** is selected, the data from it can be loaded by pressing the “Read results” button.

The saved data are separated into two types:

Input data (geometrical parameters, plasma profiles, etc) used in the current simulation. A detailed description of the input data format is given in the Appendix A.

Output results/data. The description of the output data format is given in Appendix B.

3.2 Delete selected RUN/output_file.

In order to save memory the MDSPLUS RUNs and output_files can be deleted. In order to delete a particular run, select it from the dropdown menu, mark the “READY TO DEL” checkbox, and push the “Delete Node/File” button. Note: Only owner of a MDSPLUS run (the person who saved the data) is allowed to delete it.

3.3 Preview/Change Input Parameters.

All input data used in the calculation are separated into 9 categories (Appendix A).

The first category consists of some general parameters. Two of them, discussed in chapter 3.1, are **beam** (”Beam Type”) and **shot** (“Selected Shot”). Two others are the two time values: **t1**, **t2**, which define the time interval, along which all the plasma and beam parameters are averaged (Figure 3.3.1)

The next 7 categories contain parameters which describe some global beam and plasma parameters and geometry. Each of these categories has a button (Figure 3.3.) which will open a popup window in which parameters can be viewed or changed.

The last category contain input parameters which can be viewed but not modified within the ALCBEAM widget. These define the **n_e**, **t_e** and **z_eff** profiles of the plasma and their errors.

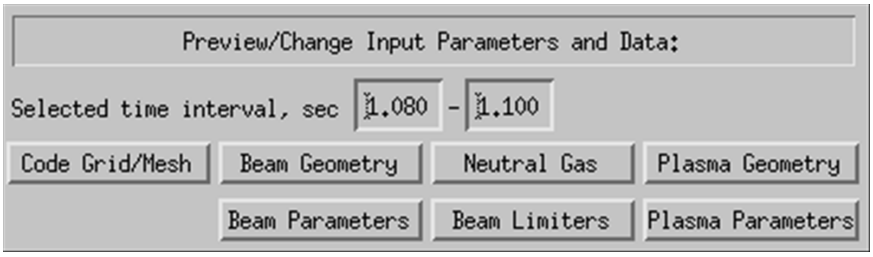


Figure 3.3.1 Input parameters

In the next 9 subsections, each of these categories is discussed in detail.

3.3.1 Beam, Shot and Selected time interval for calculation.

Two parameters discussed in chapter 3.1 are **beam** (”Beam Type”) and **shot** (“Selected Shot”). Other two are the time values: **t1**, **t2** which define the time interval used for a simulation. All parameters which depend on time will be averaged along this interval. A detailed description of these parameters can be found in (Appendix A.1).

3.3.2 Code Grid/Mesh parameters.

This category of parameters is opened with the “Code Grid/Mesh button”. A snapshot of the Code Grid/Mesh window is shown in Figure 3.3.2. There are two tables of values which define the resolution of the 3D grid/mesh used in calculations. There are 3 dimensions called (**z_beam**,

x_beam and **y_beam**). **z_beam is the direction of beam propagation**. **y_beam** is the direction parallel to the Z-axis of the Tokamak (bottom->top). **x_beam** axis is defined as a cross-product of (**z_beam** and **y_beam**) and lays in the midplane of the tokamak.

x_beam and **y_beam** are defined by 3 values: Min, Step, Max.

z_beam is defined by 2 regular intervals (Min: Step1: Mid) and (Mid: Step2: Max).

A detailed description of these parameters can be found in Appendix A.2.

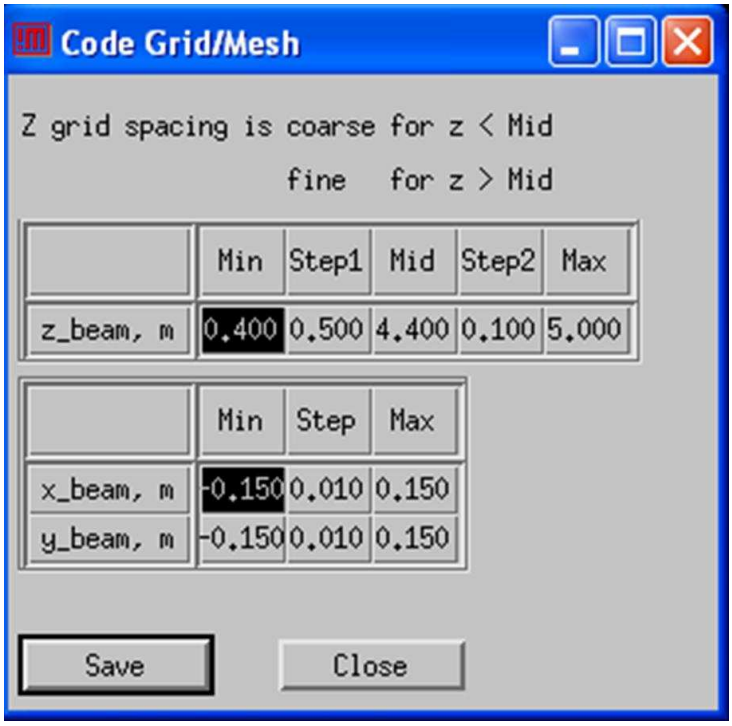


Figure 3.3.2 Code Grid/Mesh

The Z-grid has extra flexibility for defining two regions of different resolution since more accurate/fine grid calculations are needed where the beam interacts with the plasma than in the beam tank region. **User should press enter after each value of the table is changed.** The save button is used to accept the values as changed. The window is closed by the “Close” button. Unsaved changes are discarded.

3.3.3 Beam Geometry parameters.

This category of parameters is opened with the “Beam Geometry” button. Two snapshots of the Beam Geometry window are shown in Figure 3.3.3 and Figure 3.3.4. This window contains all the parameters which define the position of the Beam Injector relative to the tokamak. A detailed description of these parameters can be found in Appendix A.3. On the bottom of the window a

geometrical sketch of the beam injector and the tokamak is shown. A switch in the right/bottom corner will allow changing the view from Top View to Side View.

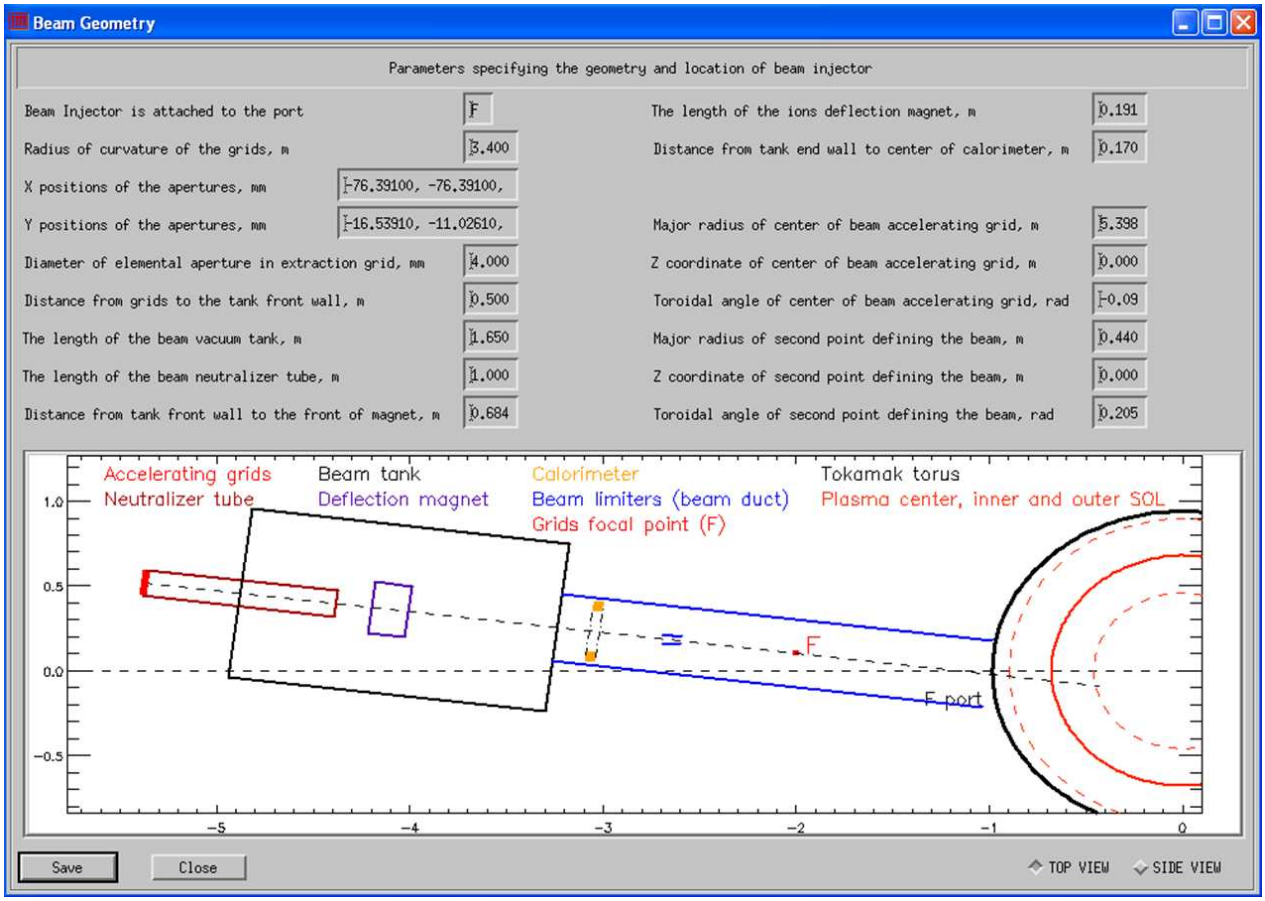


Figure 3.3.3 Beam Geometry (Top view)

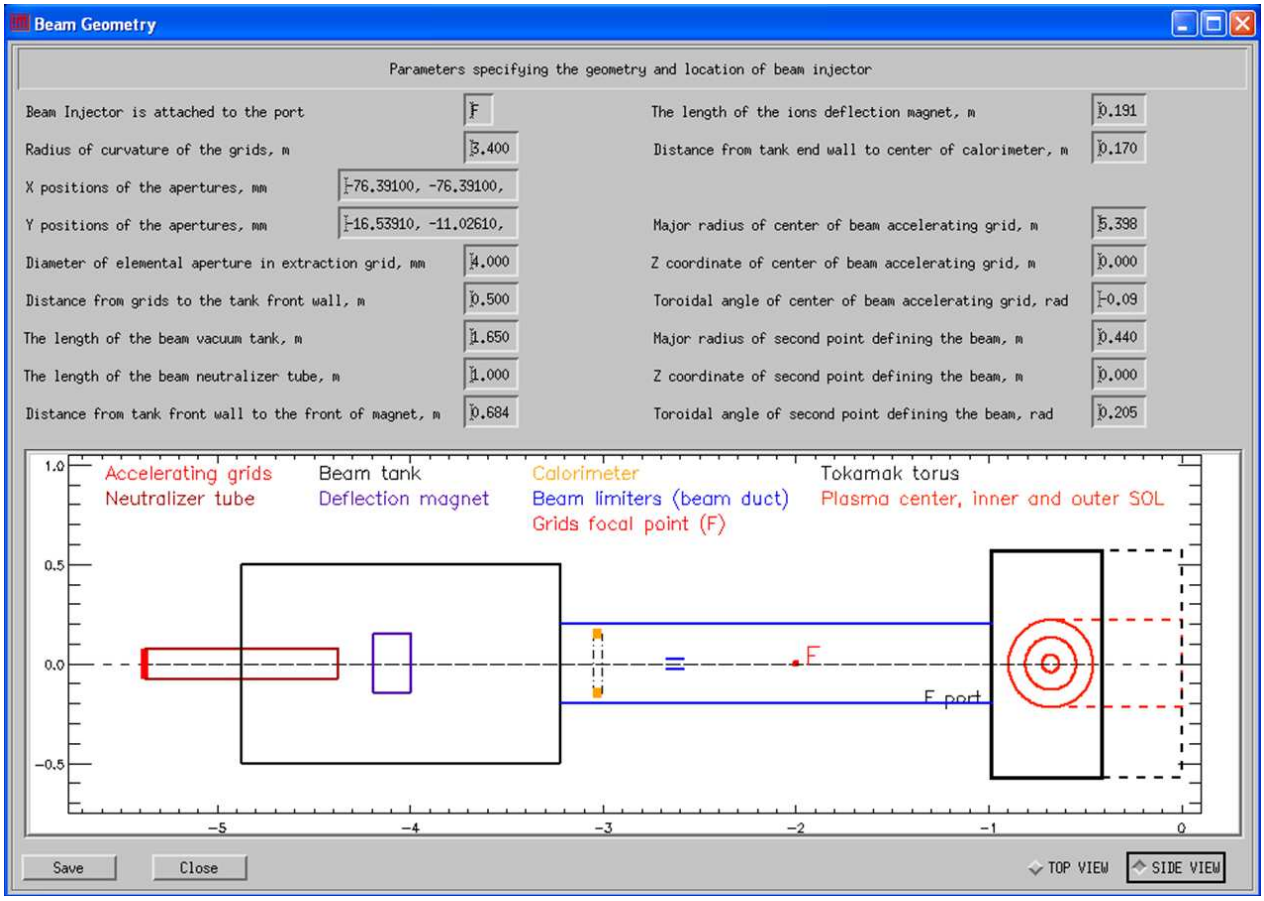


Figure 3.3.4 Beam Geometry (Side view)

The save button is used to save the values which were modified. The geometrical sketches are updated as well. The window is closed by the “Close” button. Unsaved changes are discarded.

3.3.4 Neutral Gas parameters.

This category of parameters is opened with the “Neutral Gas” button. A snapshot of the Neutral Gas window is shown in Figure 3.3.5. This window contains two parameters: one is the gas (hydrogen) pressure in the beam tank; another is the gas pressure (deuterium) in the tokamak torus. A detailed description of these parameters can be found in Appendix A.4.

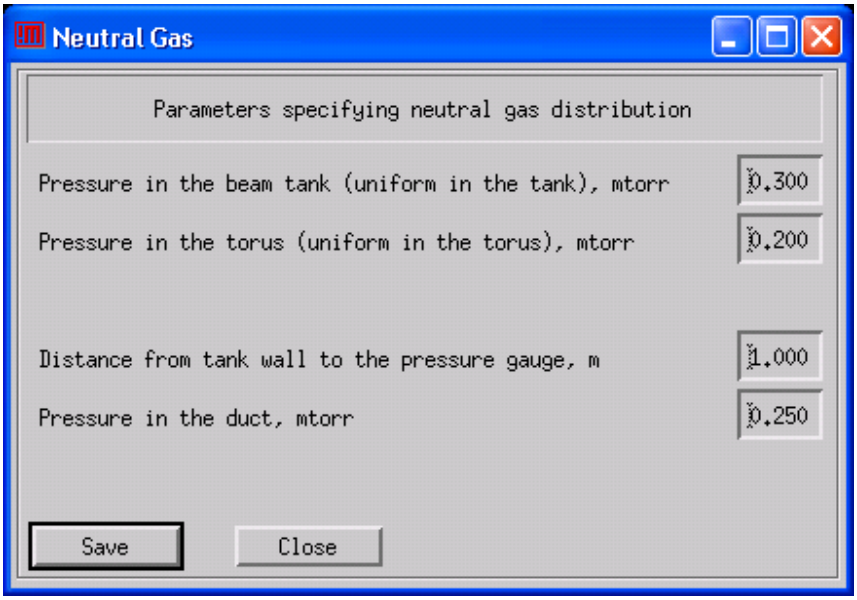


Figure 3.3.5 Neutral gas

The save button is used to accept the values as changed. The window is closed by the "Close" button. Unsaved changes are discarded.

3.3.5 Plasma Geometry parameters.

This category of parameters is opened with the "Plasma Geometry" button. A snapshot of the Plasma Geometry window is shown in Figure 3.3.6. This window contains a set of parameters which are used to construct flux surfaces of the tokamak plasma. These parameters are not used when flux surface mapping is taken from EFIT (section 3.5.2.1). A detailed description of these parameters can be found in Appendix A.5.

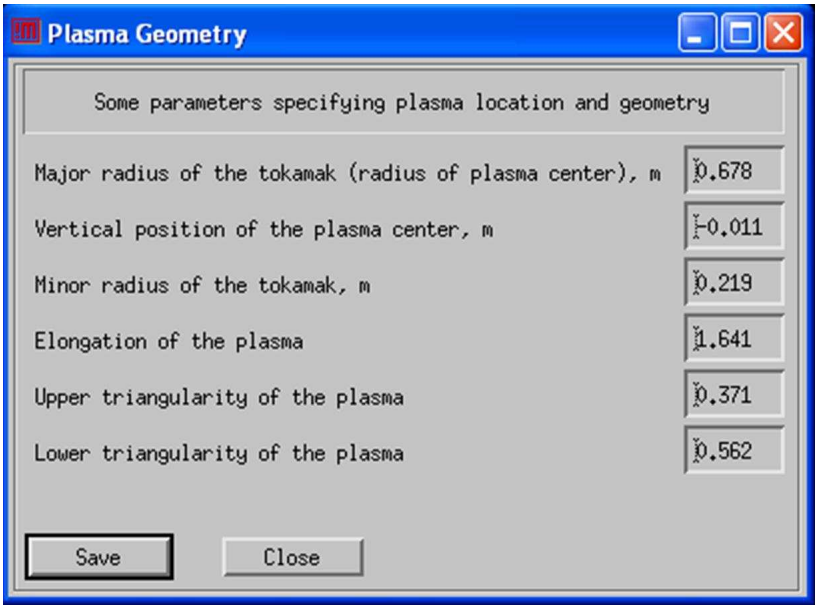


Figure 3.3.6 Plasma Geometry

The save button is used to accept the values as changed. The window is closed by the "Close" button. Unsaved changes are discarded.

3.3.6 Beam Parameters

This category of parameters is opened with the "Beam Parameters" button. A snapshot of the Beam Parameters window is shown in Figure 3.3.7. This window contains a set of parameters which describe the particle and energetic characteristics of the beam plasma source and beam itself. A detailed description of these parameters can be found in Appendix A.6.

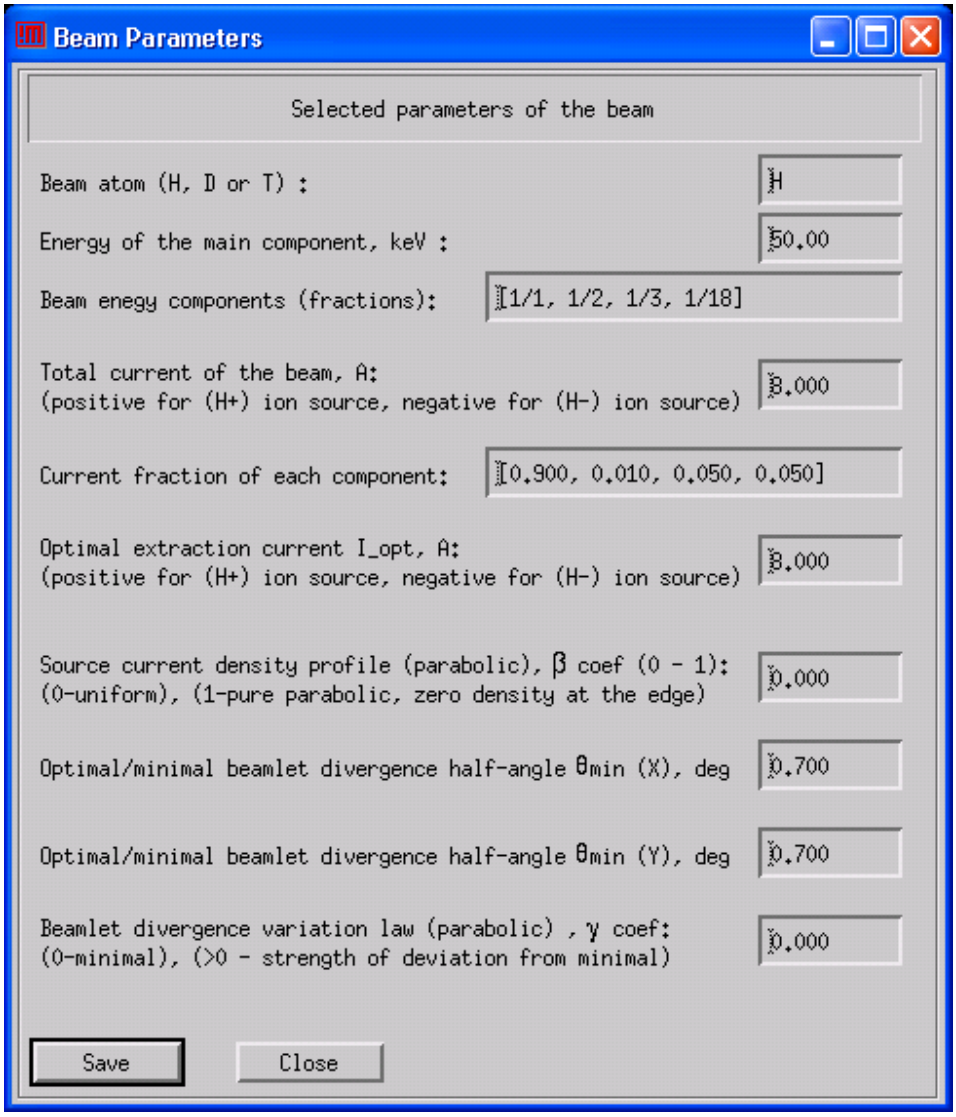


Figure 3.3.7 Beam parameters

The save button is used to accept the values as changed. The window is closed by the “Close” button. Unsaved changes are discarded.

3.3.7 Beam Limiters parameters.

This category of parameters is opened with the “Beam Limiters” button. A snapshot of the Beam Limiters window is shown in Figure 3.3.8. This window contains a set of parameters which describe the geometrical positions of the limiters/apertures located in the path of the beam propagation. There are three types of limiters/apertures can be handled by the code: circular apertures, rectangular apertures and tokamak walls. Circular apertures are defined by a name and

three geometrical parameters: (name, Z_pos, Z_size, Diameter). Rectangular apertures are defined by a name and four geometrical parameters: (name, Z_pos, Z_size, X_size, Y_size). Tokamak walls are defined by a name and one geometrical parameter: (R_major). A detailed description of these parameters can be found in Appendix A.7.

The user can add arbitrarily many apertures/limiters to the calculation. In order to do this a ”Number of apertures/limiters” should be changed and “Update” button should be pressed. This action will modify a parameters table to have a needed number of rows. Then needed parameters should be modified. Note: For circular apertures: X_size and Y_size values should be set up to “NAN”; for rectangular apertures: Diameter should be set up to “NAN”; for tokamak walls: all parameters except R_major should be set up to “NAN”. In the example in Figure 3.3.8 a beam_duct is represented by circular aperture with diameter of 0.2 m and 2.25 m long.

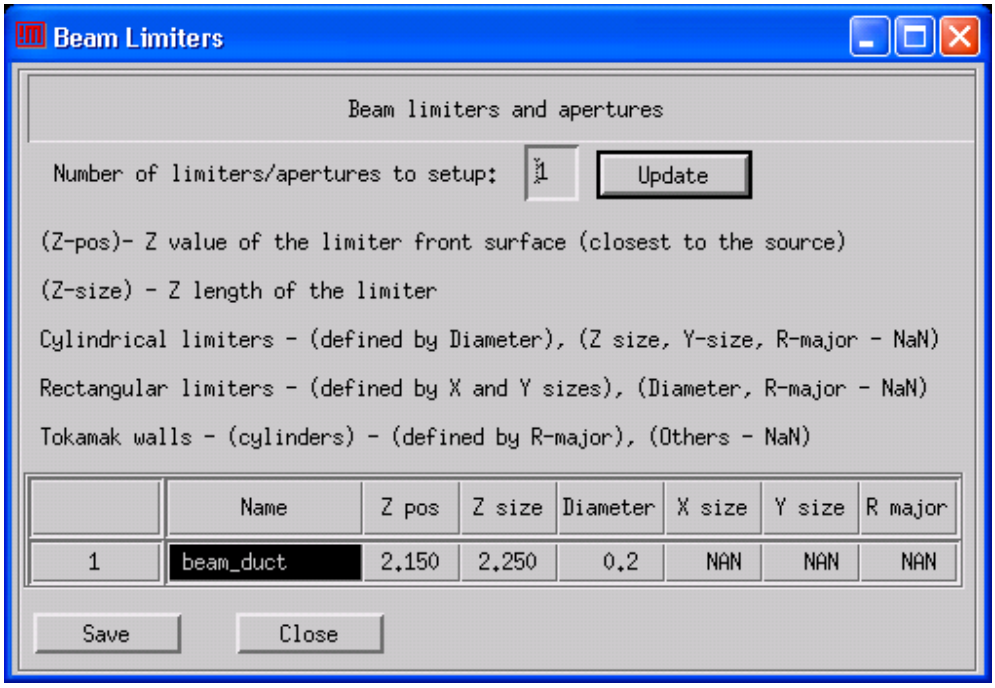


Figure 3.3.8 Beam limiters

The save button is used to accept the values as changed. The window is closed by the “Close” button. Unsaved changes are discarded.

3.3.8 Plasma parameters.

This category of parameters is opened with the “Plasma Parameters” button. A snapshot of the Plasma Parameters window is shown in Figure 3.3.9. This window contains a set of parameters which describe the ion content the plasma. The impurity density is obtained from the Z-eff

profile and the impurity density fractions. A detailed description of these parameters can be found in Appendix A.8

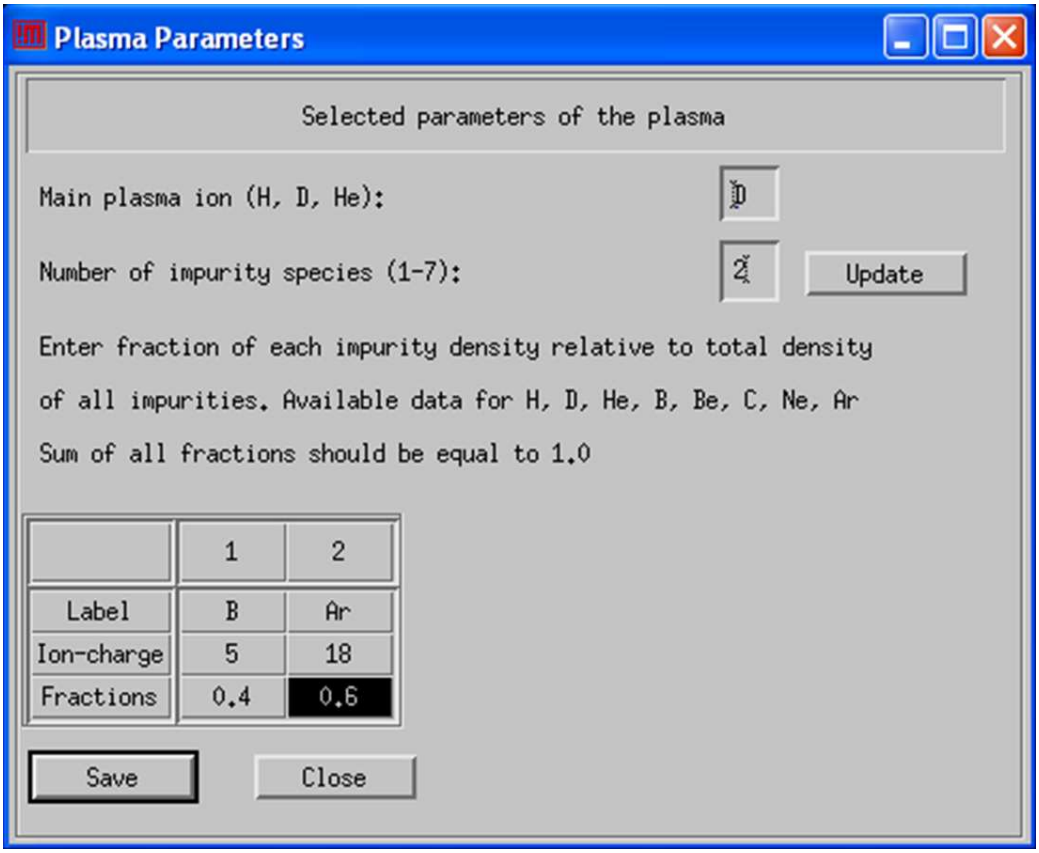


Figure 3.3.9 Plasma Parameters

3.3.9 Electron density (n_e), temperature (t_e) and plasma z_eff profiles.

This category includes plasma parameters which are needed to calculate the beam attenuation in the plasma. These parameters are: density of the electrons (**n_e**), temperature of the electrons (**t_e**), plasma effective charge (**z_eff**). A detailed description of these parameters can be found in Appendix A.9. All of these parameters are introduced as 1D profiles (value vs. major radius) of parameter itself and its errorbars. Note: errorbars are not currently used in the calculation.

3.3.10 Preview input data.

Some of the beam and plasma parameters discussed above, which are constituted as sets of data, can be previewed by the user.

3.3.10.1 Apertures in the accelerating grid.

This set of data (see Chapter 3.3.3 and Appendix A.3) are represented by two arrays of the X and Y position of the elemental apertures/holes in the last beam accelerating grid. Since there are typically about 100-1000 apertures in the beam grid, these parameters are not entered from the GUI, but instead are loaded from the MDPLUS, input (*.abi), or output (*.abo) files (see Appendix C, Appendix D, Appendix G). However, the positions of the beam apertures can be previewed by control elements in the following part of the main window (see Figure 3.3.10). "All grid apertures" value should be selected first, then press "Plot" button left of the selection. A 2D graph of the all apertures should appear in the viewer region of the main window (see Figure 3.3.11).

Figure 3.3.10 Preview input data

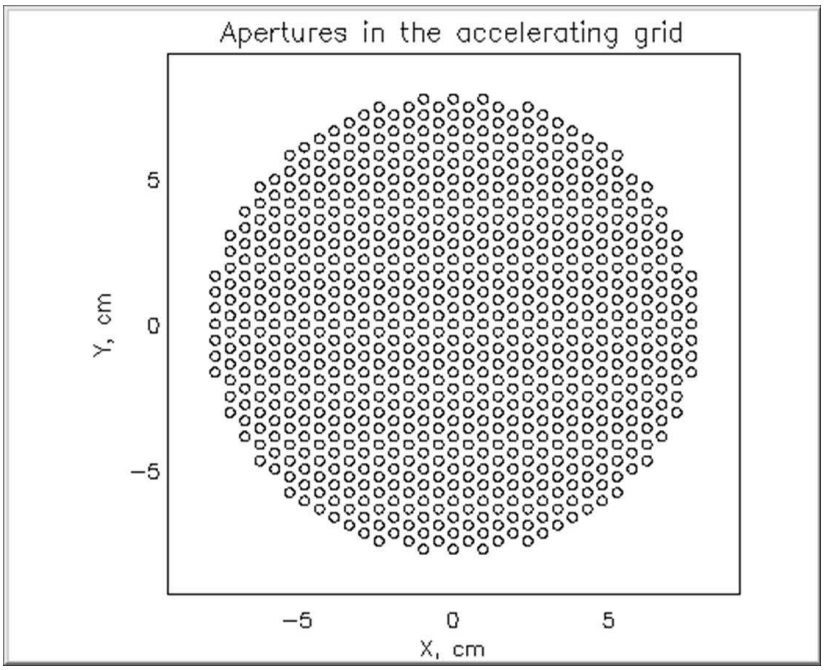


Figure 3.3.11 Preview apertures in the accelerating grid

3.3.10.2 Extraction current density distribution.

The current density distribution of the beam source plasma in front of the accelerating grid is critical parameter for proper beam extractions. The modeled shape of the distribution is defined by only one parameter: **i_dens_par** (chapter 3.3.6 or Appendix A.6). This parameter can be changed in the "Beam Parameters" window. But source plasma distribution can be previewed as a contour (Figure 3.3.12 and Figure 3.3.13). Note: The apertures are also plotted for visual comparison to their positions.

Figure 3.3.12 Preview input data

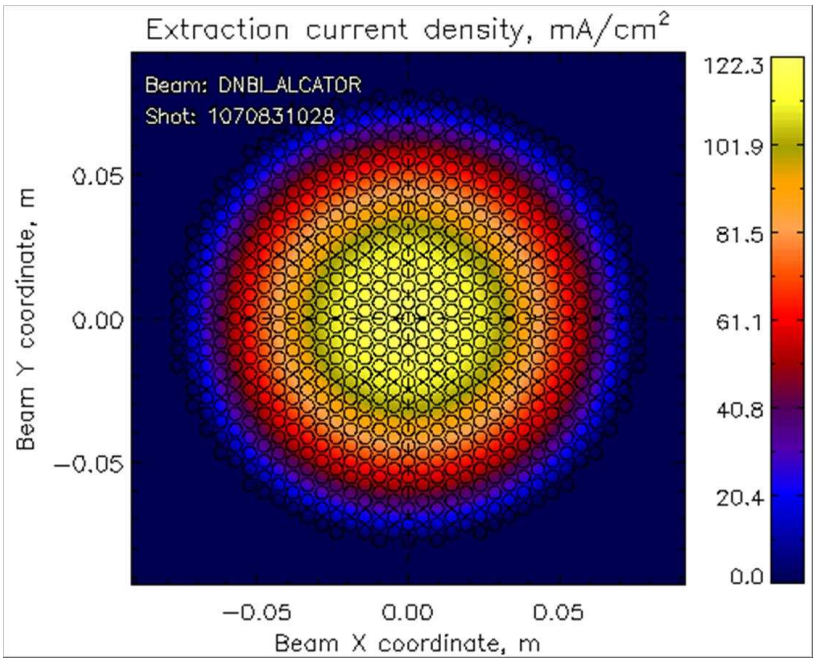


Figure 3.3.13 Preview extraction current density

3.3.10.3 Electron density (n_e), temperature (t_e) and plasma z_eff profiles.

These 3 types of plasma parameters are introduced as 1D profiles (value vs. major radius) of the parameter itself and its errorbars. They are represented by 9 arrays: **n_e_raw_r**, **n_e_raw**, **n_e_raw_err**, **t_e_raw_r**, **t_e_raw**, **t_e_raw_err**, **z_eff_raw_r**, **z_eff_raw**, **z_eff_raw_err**.

Similar to “grid apertures” these 9 arrays are typically loaded either from MDSPLUS (saved ALCBEAM RUN), input (*.abi) or output (*.abo) files (see Appendix C, Appendix D, Appendix G). However if the loaded profiles are not smooth enough (such as when they loaded directly from a diagnostic raw data node), they can be smoothed by entering a smoothing value and pressing the “Smooth” button (see Figure 3.3.14). The smooth function interpolates the **n_e_raw** (or **t_e_raw**, **z_eff_raw**) data across 200 regularly-spaced r values and computes a smoothed spline, and saves the results in the corresponding set of arrays: **n_e**, **n_e_r**, **n_e_err** (or **t_e**, **t_e_r, t_e_err**, **z_eff**, **z_eff_r**, **z_eff_err**). If smoothing is not used, the raw arrays will be copied directly into the corresponding result array. Note: the “Smooth” button only applies smoothing to the arrays of the parameter which is selected in the “Preview Droplist”.

Figure 3.3.14 Preview input profiles

Any of the **n_e**, **t_e** or **z_eff** profiles can be previewed by hitting the “Plot” button. Again, the parameter of interest should be selected in the “Preview Droplist”. An example of the preview of the **n_e** profile is shown in Figure 3.3.15. Black squares are the **n_e_raw** values. **n_e_raw** +/-**n_e_raw_err** errorbars are also plotted. Smoothed **n_e** values are shown as red curve. **n_e** +/- **n_e_err** are shown as two blue curves. Similar previews are available for **t_e** and **z_eff** profiles.

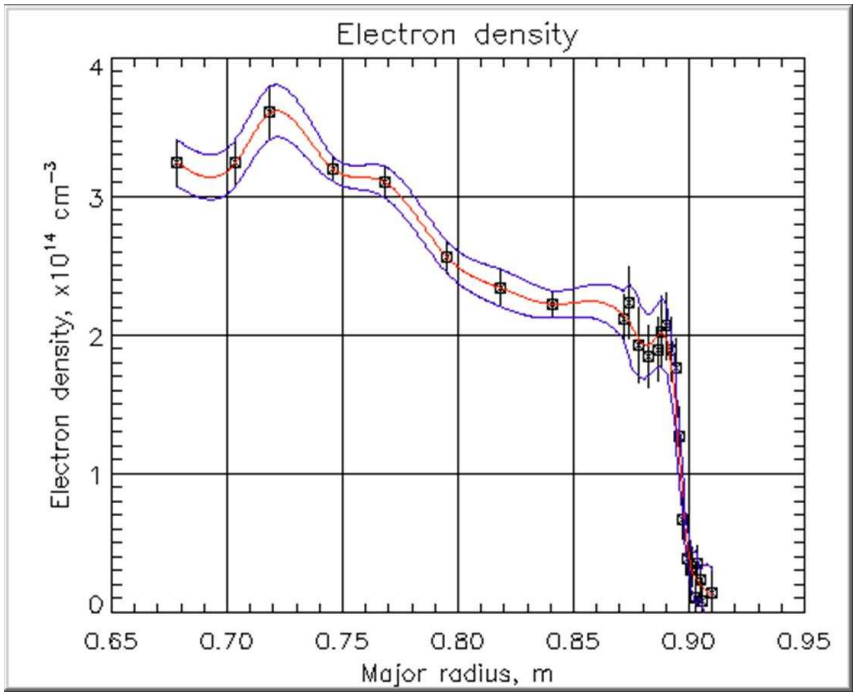


Figure 3.3.15 Preview input electron density profiles

3.3.11 Input parameters summary

All of the categories of parameters discussed above constitute the “set input parameters” for the code (described in Appendix A in details).

3.4 Plot output results.

The following area of the main window is used (Figure 3.4.1) is used to view the results of a previous ALCBEAM simulation which has been loaded (from MDSPLUS or a *.abo file) or recently run.

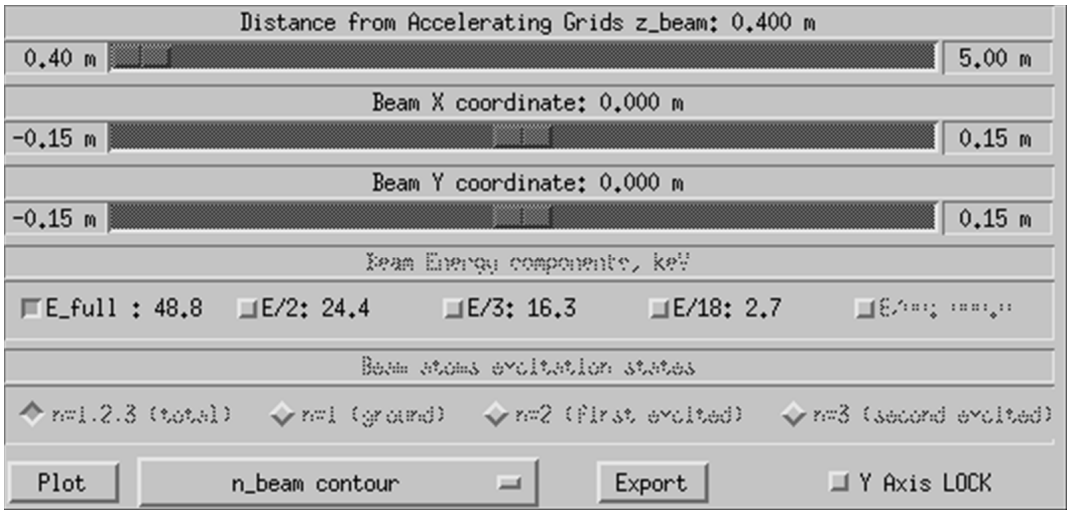


Figure 3.4.1 Widget controls for plotting output results

A detailed description of the output data/results is given in Appendix B. Here we show some examples of plotting these results. A beam density (of any beam energy component and excitation state) can be plotted in 6 different variants.

3.4.1 n_beam volume

This option plots a 3D surface of beam density as a color coded surface plot. Each of the beam cross section is colored according to the beam-line density. The user can also choose which energy component and excitation state to view (example in Figure 3.4.2)

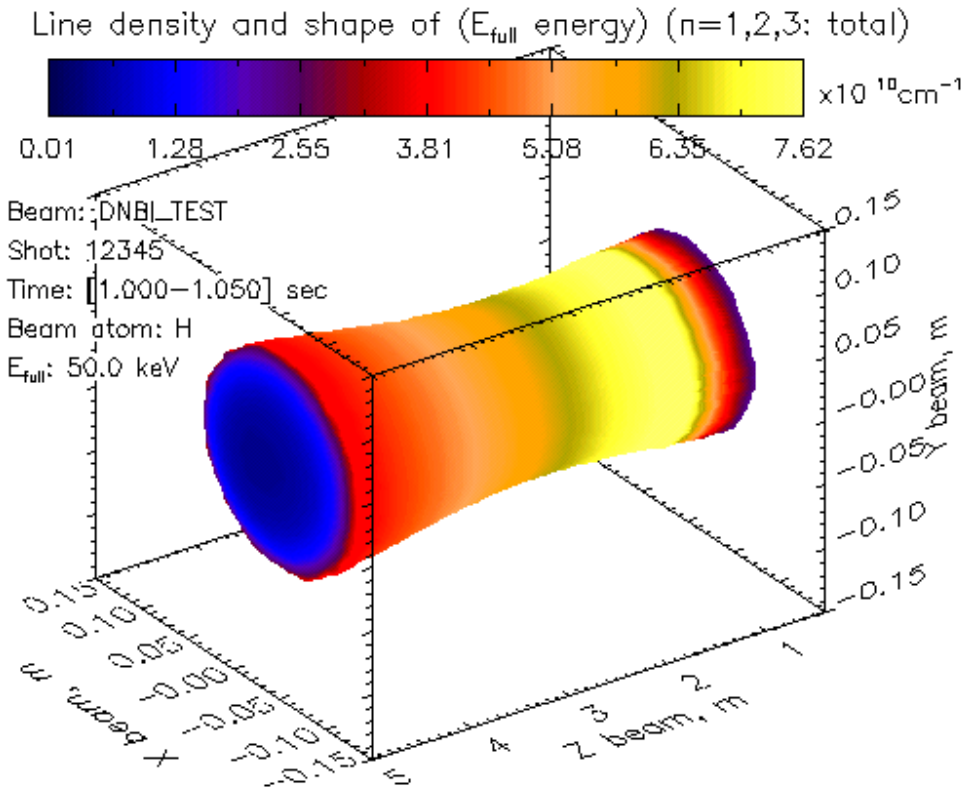


Figure 3.4.2 XY 3D surface plot of beam density from grids (E_{full} component, n=1,2,3 (total))

3.4.2 n_beam contour.

This option plots a XY cross section of the 3D array of beam density as a color contour plot. The **z_beam** slider is activated, so user can choose the **z_beam** values. The user can also choose which energy component and excitation state to view (example in Figure 3.4.3)

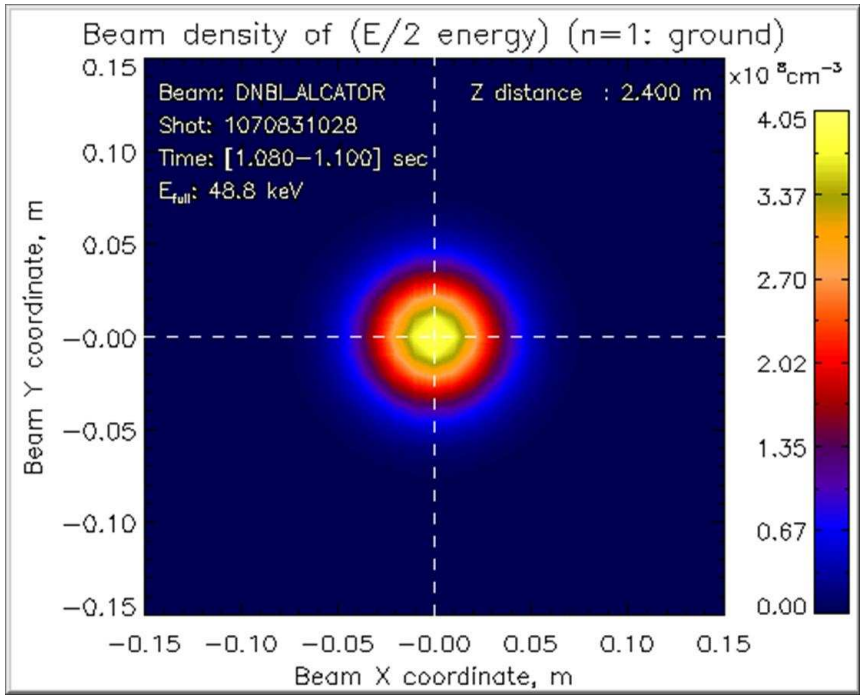


Figure 3.4.3 XY Contour plot of beam density @ z_beam=2.4 m from grids (E/2 component, n=1 (ground state))

If "Y Axis LOCK" checkbox is selected, the color range (MIN and MAX values) of the contour is frozen to the min and max values of **n_beam** in the whole range of **z_beam**. If checkbox is not selected, the color range (MIN and MAX values) of the contour is recalculated based on the min and max values of the currently selected 2D slice.

3.4.3 beam line density vs z_beam

This option plots the line density of beam atoms for each beam energy component against a distance from the accelerating grids (**n_beam_linear** [1/cm] vs. **z_beam**). Line density is supposed to be preserved (along the **z_beam**) if no attenuation is included into the calculation. The user can also choose several energy components and excitation states to plot (example in Figure 3.4.4).

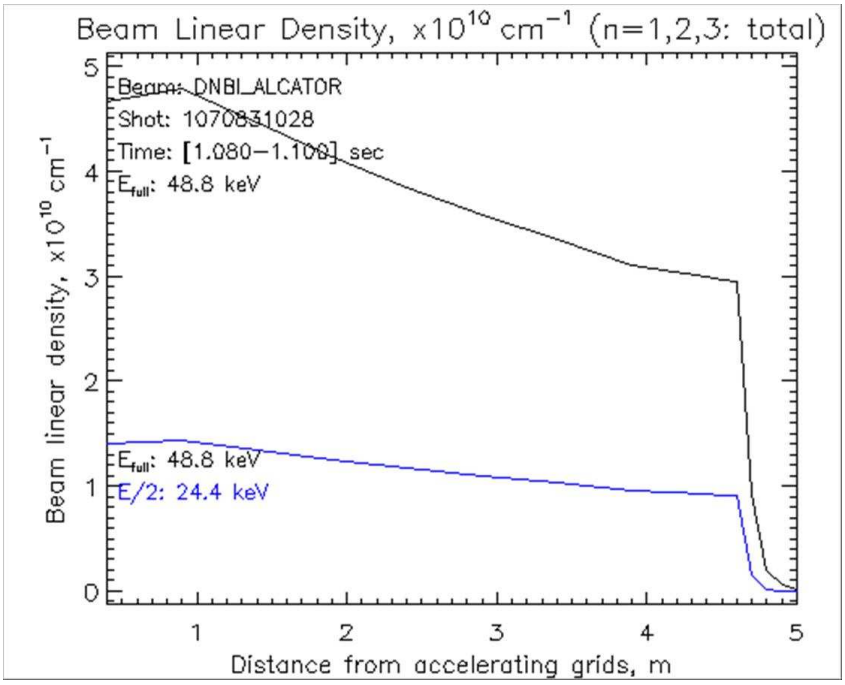


Figure 3.4.4 Beam linear density vs. z_beam (E_{full} and E/2 component, all atoms (in ground and excited states)

3.4.4 beam line density vs r_major

This option plots the line density of beam atoms for each beam energy component against tokamak major radius (**n_beam_linear** [1/cm] vs. **r_major**). The user can also choose several energy components and excitation states to plot (example in Figure 3.4.5).

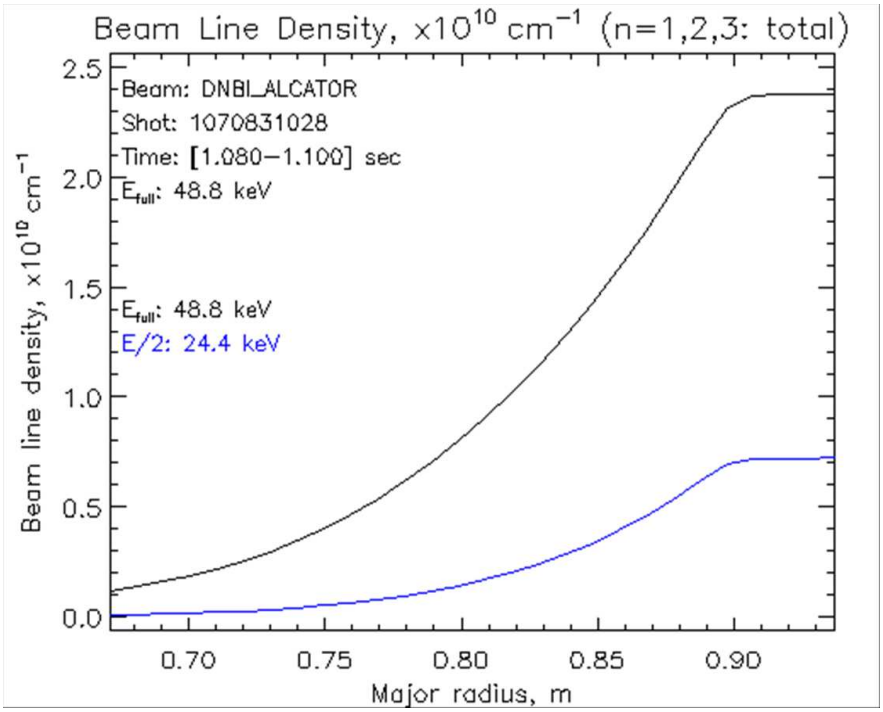


Figure 3.4.5 Beam linear density vs. r_major (E_{full} and E/2 component, all atoms (in ground and excited states)

3.4.5 beam deposition vs rho

Plot of flux surface averaged beam deposition vs rho was used to compare ALCBEAM with 2D output of NUBEAM. Can only be used after run was performed. Cannot be used when output data was read out from the output file or MDSPLUS.

3.4.6 total beam power

This option plots the total beam power of the neutral beam (**beam_power** [W] vs. **z_beam**). Beam power is supposed to be preserved (along the **z_beam**) if no attenuation is included into the calculation (example in Figure 3.4.6).

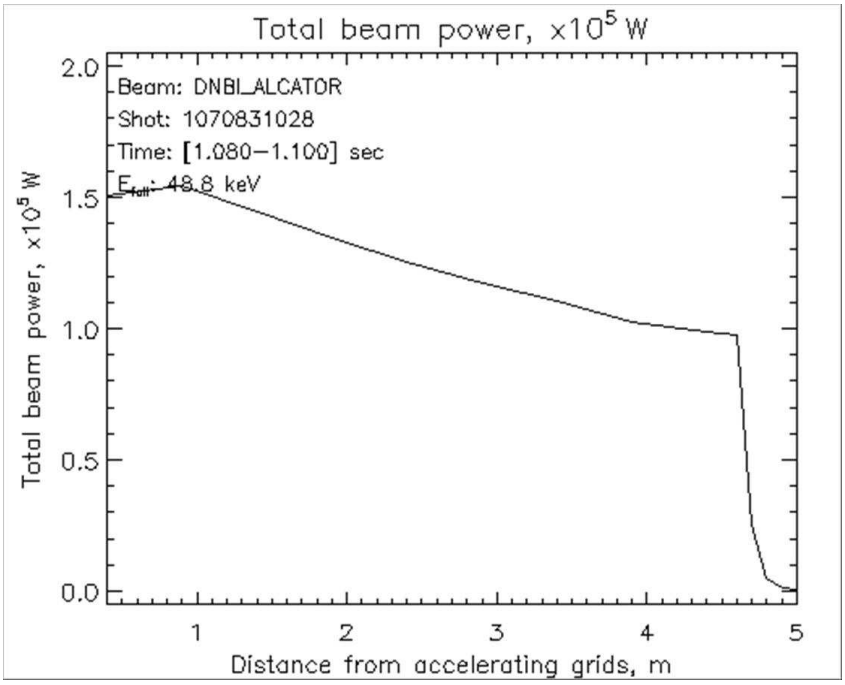


Figure 3.4.6 Total beam power vs. z_beam

3.4.7 beam horizontal width.

This option plots the FWHM of the beam in the horizontal (**x_beam**) direction at **y_beam**=0 (**x_FWHM** vs. **z_beam**). FWHM is chosen as a measure of the width since it can be calculated for any kind of beam shape and can be easily interpreted in case of Gaussian beam. The user can also choose several energy components and one excitation state to plot (example in Figure 3.4.7).

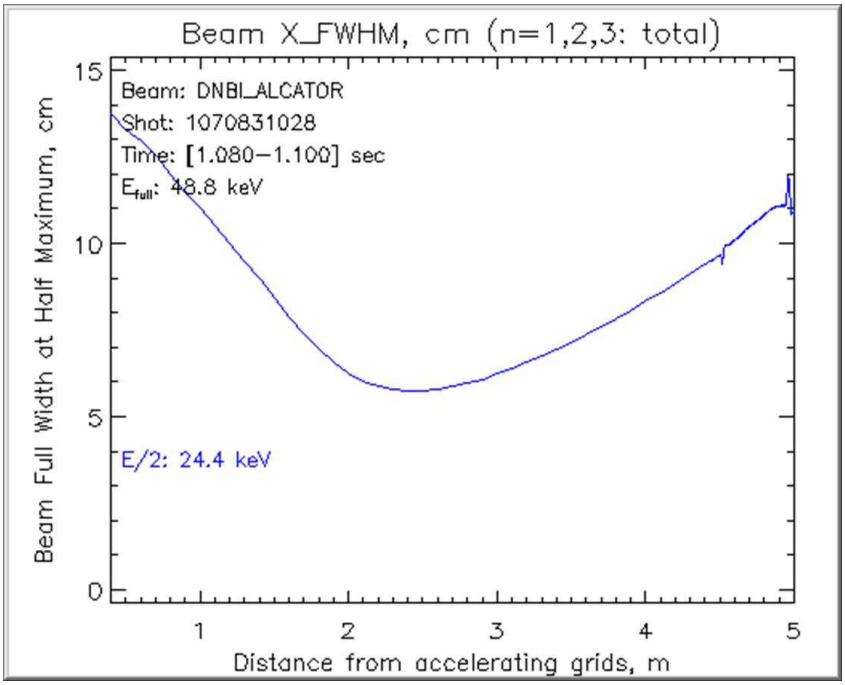


Figure 3.4.7 Beam X_FWHM of density profile vs. z_beam (y_beam=0) (E/2 component, all atoms (in ground and excited states)

3.4.8 beam vertical width.

This option plots the FWHM of the beam in vertical (**y_beam**) direction at **x_beam**=0 (**y_FWHM** vs. **z_beam**). FWHM is chosen as a measure of the width since it can be calculated for any kind of beam shape and can be easily interpreted in case of Gaussian beam. The user can also choose several energy components and one excitation state to plot (example in Figure 3.4.8).

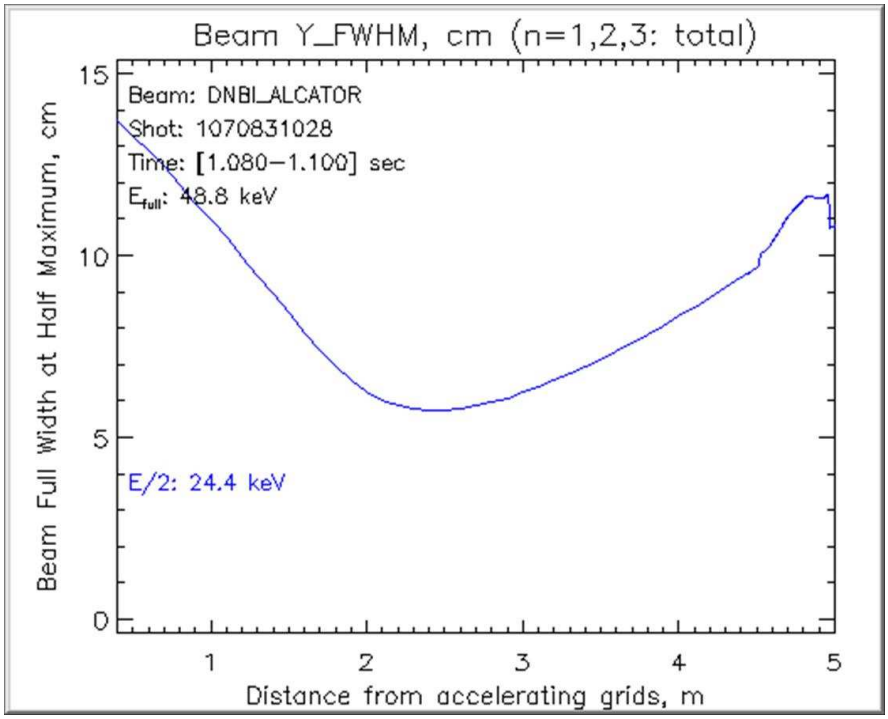


Figure 3.4.8 Beam Y_FWHM of density profile vs. z_beam (X_beam=0) (E/2 component, all atoms (in ground and excited states)

3.4.9 n_beam vs. z_beam.

This option plots **n_beam** vs. **z_beam**. **x_beam** and **y_beam** sliders are active and user can select which **x_beam** and **y_beam** to plot. The user can also choose several energy components and one excitation state to plot (example in Figure 3.4.9).

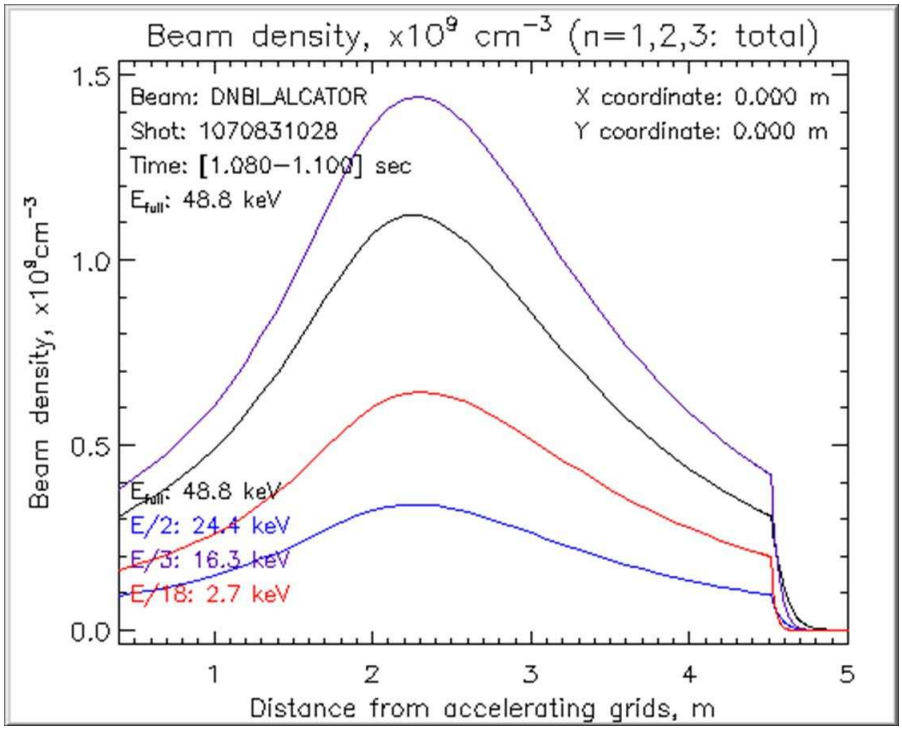


Figure 3.4.9 Example of n_beam vs z_beam plot (x_beam=0.0, y_beam=0.0). (All 4 beam energy component are shown in different colors, density of atoms in all atomic states are plotted (ground and excited states)

If "Y Axis LOCK" checkbox is selected, the (MIN and MAX values) of the graph are frozen to the min and max values of **n_beam** in the whole 3D range of **x_beam**, **y_beam** and **z_beam**. If checkbox is not selected, the (MIN and MAX values) of the graph are recalculated based on the min and max values of 1D **n_beam** of this particular **x_beam** and **y_beam** (selected by the slider).

3.4.10 n_beam vs. r_major.

This option plots **n_beam** vs. tokamak major radius. **x_beam** and **y_beam** sliders are active and user can select which **x_beam** and **y_beam** to plot. The user can also choose several energy components and one excitation state to plot (example in Figure 3.4.9).

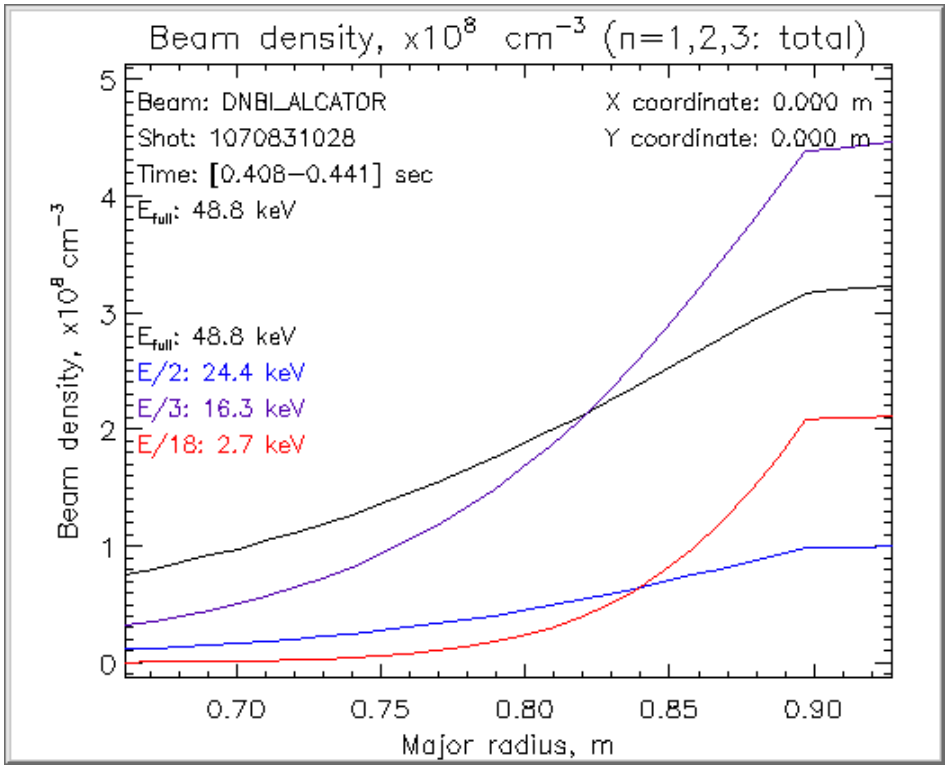


Figure 3.4.10 Example of n_beam vs r_major plot (x_beam=0.0, y_beam=0.0). (All 4 beam energy component are shown in different colors, density of atoms in all atomic states are plotted (ground and excited states)

If "Y Axis LOCK" checkbox is selected, the (MIN and MAX values) of the graph are frozen to the min and max values of **n_beam** in the whole 3D range of **x_beam**, **y_beam** and **z_beam**. If checkbox is not selected, the (MIN and MAX values) of the graph are recalculated based on the min and max values of 1D **n_beam** of this particular **x_beam** and **y_beam** (selected by the slider).

3.4.11 n_beam vs. x_beam.

This option plots **n_beam** vs. **x_beam**. **z_beam** and **y_beam** sliders are active and user can select which **z_beam** and **y_beam** will be used for plotting. The user can also choose several energy components and one excitation state to plot (example in Figure 3.4.11). Asymmetry of the beam profile is calculated as:

Xmax=max(x_beam) Xmin=min(x_beam)

x1=Xmin/3.0 x2=Xmax/3.0

y=y_beam slider position z=z_beam slider position

Asymmetry=(n(x2,y,z))-n(x1,y,z))/(n(x1,y,z)+n(x2,y,z))*200.0 ; in percents

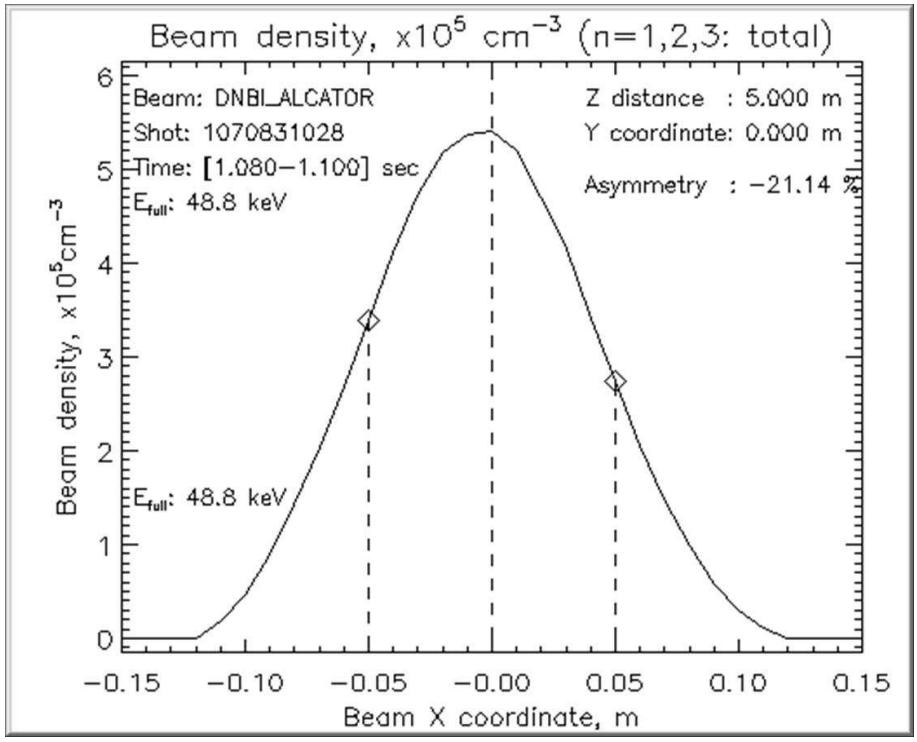


Figure 3.4.11 Example of n_beam vs x_beam plot (z_beam=5.0, y_beam=0.0). (E_full energy component is shown, density of atoms in all atomic states are plotted (ground and excited states)

If "Y Axis LOCK" checkbox is selected, the (MIN and MAX values) of the graph are frozen to the min and max values of **n_beam** in the whole 3D range of **x_beam**, **y_beam** and **z_beam**. If checkbox is not selected, the (MIN and MAX values) of the graph are recalculated based on the min and max values of 1D **n_beam** of this particular **y_beam** and **z_beam** (selected by the slider).

3.4.12 n_beam vs. y_beam.

This option plots **n_beam** vs. **y_beam**. **z_beam** and **x_beam** sliders are active and user can select which **z_beam** and **x_beam** will be used for plotting. The user can also choose several energy components and one excitation state to plot (example in Figure 3.4.12). Asymmetry of the beam profile is calculated as:

Ymax=max(y_beam) Ymin=min(y_beam)

x1=Ymin/3.0 x2=Ymax/3.0

x=x_beam slider position z=z_beam slider position

Asymmetry=(n(x,y2,z)-n(x,y1,z))/(n(x,y1,z)+n(x,y2,z))*200.0 ; in percents

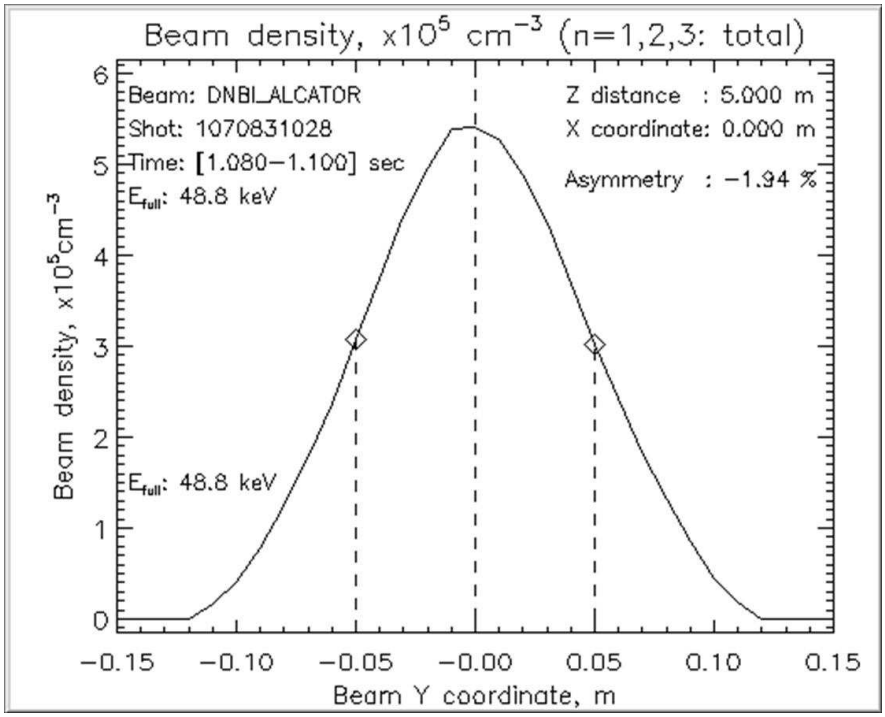


Figure 3.4.12 Example of n_beam vs y_beam plot (z_beam=5.0, x_beam=0.0). (E_full energy component is shown, density of atoms in all atomic states are plotted (ground and excited states)

If "Y Axis LOCK" checkbox is selected, the (MIN and MAX values) of the graph are frozen to the min and max values of **n_beam** in the whole 3D range of **x_beam**, **y_beam** and **z_beam**. If checkbox is not selected, the (MIN and MAX values) of the graph are recalculated based on the min and max values of 1D **n_beam** of this particular z**_beam** and **x_beam** (selected by the slider).

3.4.13 lost atoms line density

This option plots the linear density of beam atoms lost from the beam due to the interaction with gas, plasma or limiters (**n_beam_lost** [1/cm^2] vs. **z_beam**). This parameter defined as an absolute fraction of linear density which was lost on the 1 cm of the beam path. It used to calculate the beam power loss (see chapter 3.4.14). The user can also choose several energy components to plot (example in Figure 3.4.13).

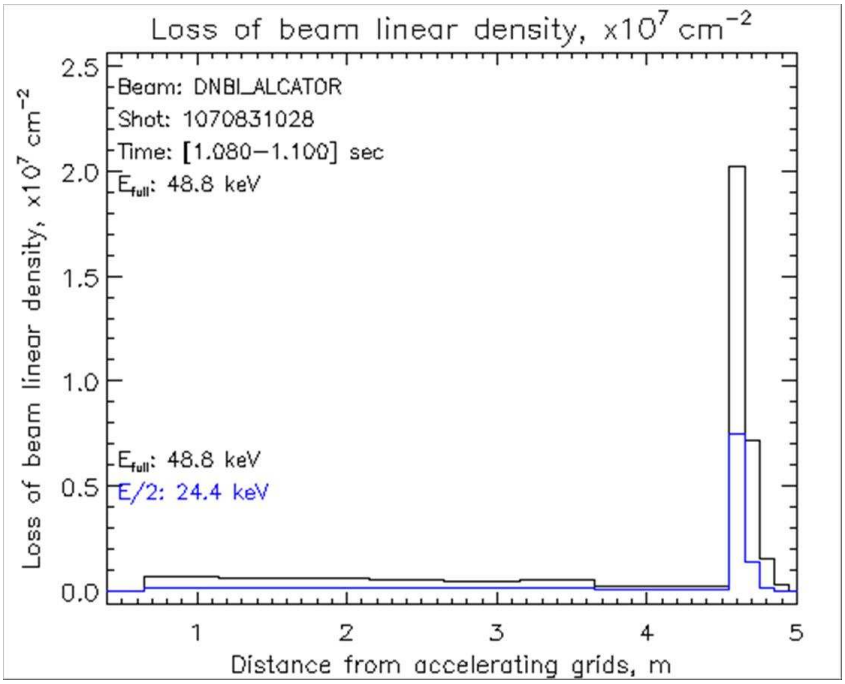


Figure 3.4.13 Linear density of atoms lost from the beam per 1 cm of the beam path: n_beam_lost vs. z_beam (E_{full} and E/2 component, all atoms

3.4.14 beam power loss

This option plots the total beam power loss per cm of beam path due to the interaction with gas, plasma or limiters (**beam_power_lost** [1/cm] vs. **z_beam**). This parameter defined how much of the beam total power was lost on 1 cm of the beam path (example in Figure 3.4.14).

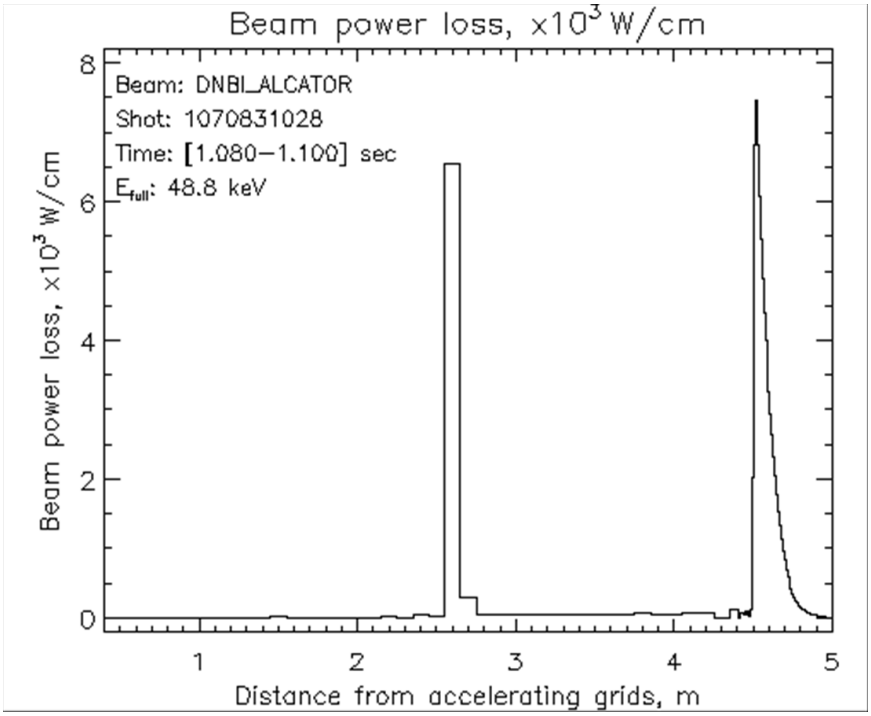


Figure 3.4.14 Beam power loss: beam_power_lost vs. z_beam (all neutrals)

3.4.15 velocity contour

This option plots a 2D array of velocity distribution at each given point of **z_beam**, **x_beam** and **y_beam**. The **z_beam**, **x_beam** and **y_beam** sliders are activated, so user can choose any **z_beam**, **x_beam** and **y_beam** values to plot. The user can also choose which energy component to view (example in Figure 3.4.15)

Velocity distribution is represented by a 2D array of velocity vectors with a given/color coded amplitude. Vector X, Y coordinates are given on the plot. Velocity Z coordinate is equal to **z_beam**. More explanation can be found in Appendix B.

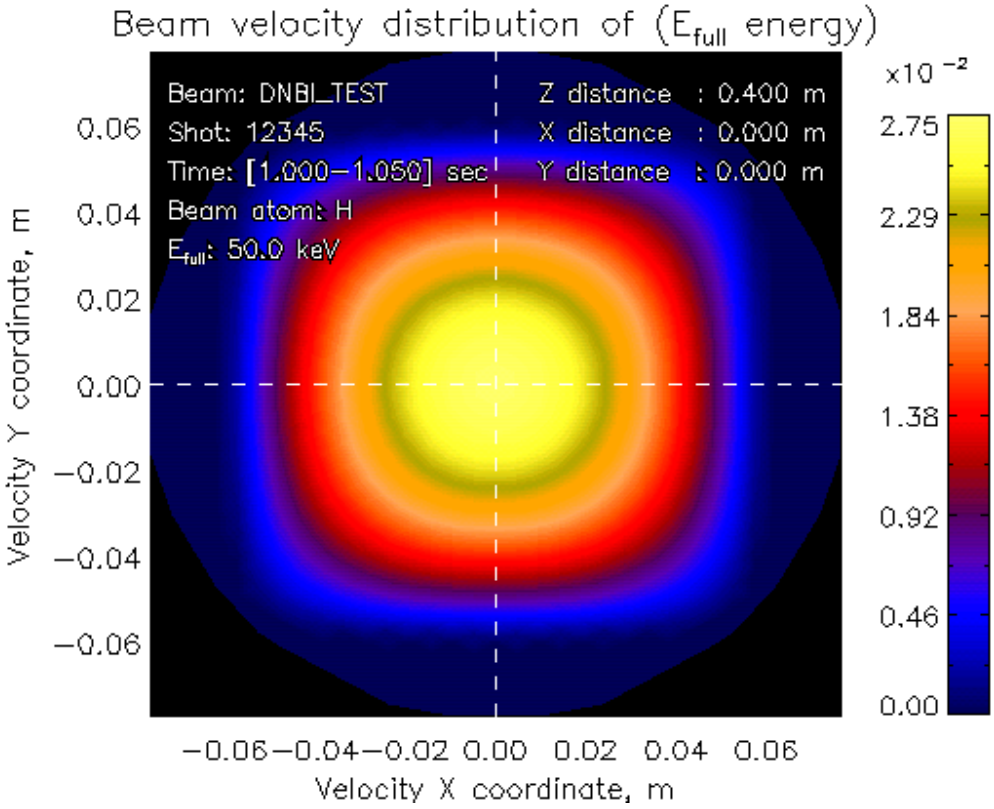


Figure 3.4.15 contour plot of the beam velocity distribution at a given z_beam, x_beam and y_beam.

If “Y Axis LOCK” checkbox is selected, the color range (MIN and MAX values) of the contour is frozen to the min and max values of peak of the **velocity distribution** in the whole range of a selected **x_beam** and **y_beam**. If checkbox is not selected, the color range (MIN and MAX values) of the contour is recalculated based on the min and max values of the currently selected 2D slice.

3.5 Run the simulation.

In this chapter is a detailed explanation of how to prepare input data, run a simulation, and save the output results.

The area of the main window which contains widget elements designed to run the code is shown in Figure 3.5.1.

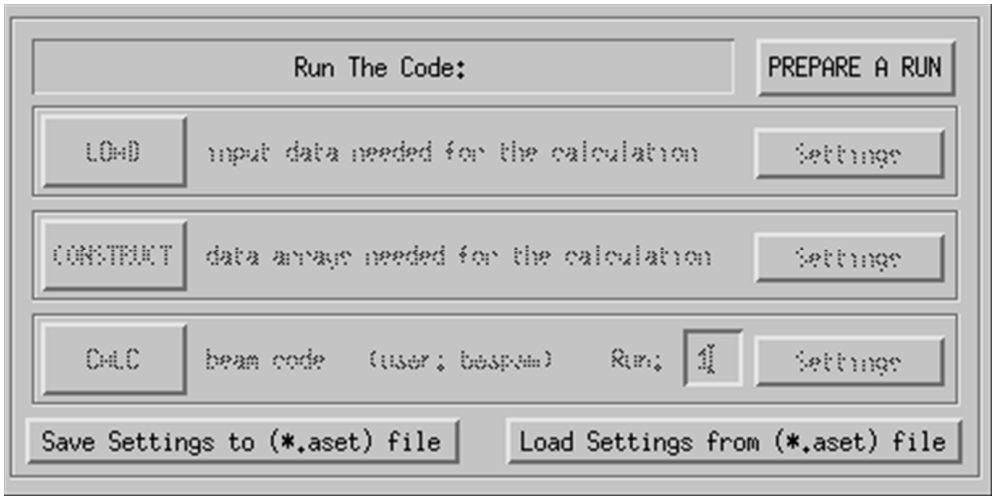


Figure 3.5.1 Run the code area in the main widget

By default this area is set up to be not sensitive, in order to prevent a user who is just reviewing saved results from pressing some buttons by mistake.

A simulation is run in two steps. First, the run is prepared (all input data is to be loaded, all supporting array are to be constructed). The next step is the actual calculation of the beam divergence and penetration.

Let's discuss each step in detail.

In order to start a new run, first press "Prepare A Run" button. This action switches the code from "PREVIEW" phase to the "RUN" phase. The label on the same button is changed to "Cancel Run" (See Figure 3.5.2). Note: If user wants to cancel the run preparation and return into "Preview" phase, this button should be pressed one more time.

After this initialization, the area "LOAD input data needed for calculation" becomes sensitive (see Figure 3.5.2).

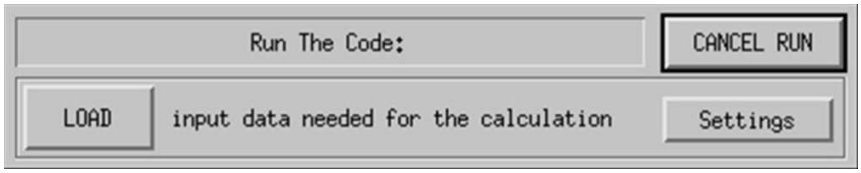


Figure 3.5.2 Load input data needed for calculation

3.5.1 Load input parameters and data.

The “LOAD” button loads the input parameters into the code environment. Before that the “Settings” button should be pressed, which opens an additional window, where the user can select the sources from which the input data should be loaded. A snapshot of this window is shown in Figure 3.5.3. In this window user previews the control elements of the same parameters discussed in chapter 3.3, but separated in different 11 subcategories. Let’s discuss them one by one:

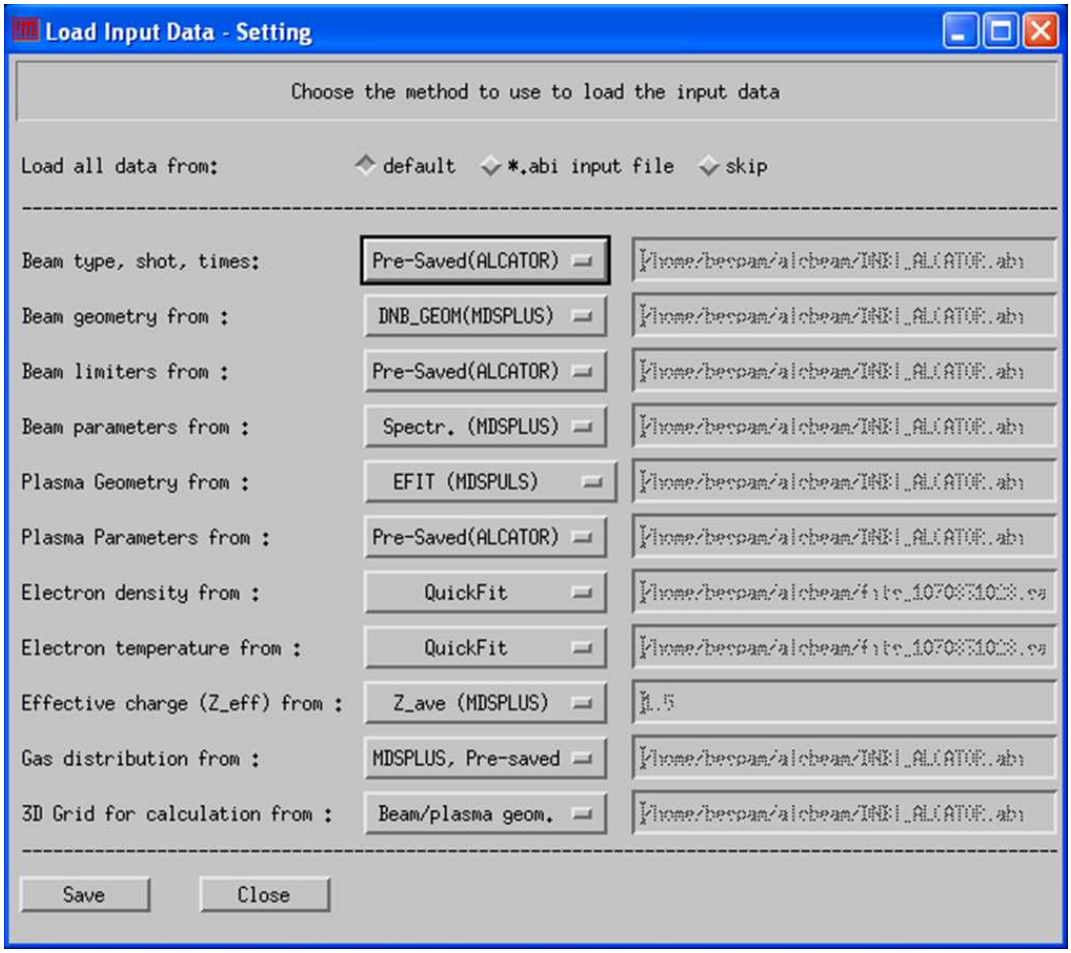


Figure 3.5.3 Load input data - Settings

3.5.1.1 Beam type, shot, times.

These parameters are discussed in chapter 3.3.1 and Appendix A.1.

There are three load options:

a) Pre-Saved (ALCATOR). This option will use values that are saved in the body of the ALCBEAM code. These values are selected for one of the default DNBI_ALCATOR runs:

beam='DNBI_ALCATOR'

shot='1070831028'

t1=1.080

t2=1.100

b) *.abi input file. This option means that the values will be loaded from *.abi input file, which name can be chosen in the text field on the right side.
c) Skip. This option means that currently loaded/displayed values of the parameters will be used. No load procedure for these parameters will be performed.

3.5.1.2 Beam geometry.

These parameters are discussed in chapter 3.3.3 and Appendix A.3.
There are three load options:

a) DNB_GEOM (MDSPLUS). This option means that parameters in this category will be loaded from the DNB_GEOM tree. This is the default option for ALCBEAM. These parameters are not listed here, but user can preview them by the methods discussed in chapter 3.3.
b) *.abi input file. This option means that the values will be loaded from *.abi input file, which name can be chosen in the text field on the right side.
c) Skip. This option means that currently loaded/displayed values of the parameters will be used. No load procedure for these parameters will be performed.

3.5.1.3 Beam limiters.

These parameters are discussed in chapter 3.3.7 and Appendix A.7.
There are three load options:

a) Pre-Saved (ALCATOR). This option will use values that are saved in the body of the ALCBEAM code. These values are selected for one of the default DNBI_ALCATOR runs. These parameters are not listed here, but user can preview them by the methods discussed in chapter 3.3.
b) *.abi input file. This option means that the values will be loaded from *.abi input file, which name can be chosen in the text field on the right side.
c) Skip. This option means that currently loaded/displayed values of the parameters will be used. No load procedure for these parameters will be performed.

3.5.1.4 Beam parameters.

These parameters are discussed in chapter 3.3.6 and Appendix A.6.

There are three load options:

a) Spectr. (MDSPLUS). This option means that parameters in this category will be loaded from the Spectroscopy tree of the ALCATOR MDSPLUS.
b) *.abi input file. This option means that the values will be loaded from *.abi input file, which name can be chosen in the text field on the right side.
c) Skip. This option means that currently loaded/displayed values of the parameters will be used. No load procedure for these parameters will be performed.

3.5.1.5 Plasma Geometry.

These parameters are discussed in chapter 3.3.5 and Appendix A.5.

There are three load options:

a) Spectr. (MDSPLUS). This option means that parameters in this category will be loaded from the Spectroscopy tree of the ALCATOR MDSPLUS.
b) *.abi input file. This option means that the values will be loaded from *.abi input file, which name can be chosen in the text field on the right side.
c) Skip. This option means that currently loaded/displayed values of the parameters will be used. No load procedure for these parameters will be performed.

3.5.1.6 Plasma parameters.

These parameters are discussed in chapter 3.3.8 and Appendix A.8.

There are three load options:

a) Pre-Saved (ALCATOR). This option will use values that are saved in the body of the ALCBEAM code. These values are selected for one of the default DNBI_ALCATOR runs. These parameters are not listed here, but user can preview them by the methods discussed in chapter 3.3.
b) *.abi input file. This option means that the values will be loaded from *.abi input file, which name can be chosen in the text field on the right side.

c) Skip. This option means that currently loaded/displayed values of the parameters will be used. No load procedure for these parameters will be performed.

3.5.1.7 Electron Density.

These input profiles are discussed in chapter 3.3.9 and Appendix A.9.

There are five load options:

a) QuickFit. This option means that **n_e** profiles will be calculated by running the QuickFit routine. This is one of the new fits for TS data available at C-Mod.

b) Thompson (MDSPLUS). This option means that **n_e** profiles will be loaded from the TS raw data node in MDSPLUS tree of ALCATOR tokamak.

c) FITS file. This option means that **n_e** profiles will be loaded from the output file of the FITS routine (smoothing routine for TS data). The filename can be chosen in the text field on the right side.

d) *.abi input file. This option means that **n_e** profiles will be loaded from *.abi input file, which name can be chosen in the text field on the right side.

e) Skip. This option means that currently loaded/displayed **n_e** profiles will be used. No load procedure for this category will be performed.

3.5.1.8 Electron Temperature.

These input profiles are discussed in chapter 3.3.9 and Appendix A.9.

There are five load options:

a) QuickFit. This option means that **t_e** profiles will be calculated by running the QuickFit routine. This is one of the new fits for TS data of ALCATOR tokamak.

a) Thompson (MDSPLUS). This option means that **t_e** profiles will be loaded from the TS raw data node in MDSPLUS tree of ALCATOR tokamak.

b) ECE (MDSPLUS). This option means that **t_e** profiles will be loaded from the ECE raw data node in MDSPLUS tree of ALCATOR tokamak.

c) FITS file. This option means that **t_e** profiles will be loaded from the output file of the FITS routine (smoothing routine for TS data). The filename can be chosen in the text field on the right side.
d) *.abi input file. This option means that **t_e** profiles will be loaded from *.abi input file, which name can be chosen in the text field on the right side.
e) Skip. This option means that currently loaded/displayed t_e profiles will be used. No load procedure for this category will be performed.

3.5.1.9 Effective Charge (Z_eff).

These input profiles are discussed in chapter 3.3.9 and Appendix A.9.
There are five load options:
a) Dalsa (MDSPLUS). This option means that **z_eff** profiles will be loaded from the DALSA output node in MDSPLUS tree of ALCATOR tokamak.
b) Z_Ave (MDSPLUS). This option means that average **z_eff** from z_ave monitor (ALCATOR) will be used to build uniform z_eff profiles.
c) Constant. This option means tha**t z_eff** is set to a constant value in plasma. The value of **z_eff** is set up in the text filed on the right side.
d) *.abi input file. This option means that **z_eff** profiles will be loaded from *.abi input file, which name can be chosen in the text field on the right side.
e) Skip. This option means that currently loaded/displayed **z_eff** profiles will be used. No load procedure for this category will be performed.

3.5.1.10 Gas distribution

These input profiles are discussed in chapter 3.3.4 and Appendix A.4.
There are three load options:
a) MDSPLUS, Pre-Saved. This option means that there are some values taken from the body of the ALCBEAM code, and some parameters are read from MDSPLUS (ALCATOR) (for example: \edge::top.gas.ratiomatic:F_side). These values are selected for one of the default DNBI_ALCATOR runs. The parameters are: **tank_pressure** (beam tank

pressure), **torus_pressure** (tokamak torus pressure), **duct_pressure** (pressure in the beam duct) and **duct_pressure_loc** (distance from the nearest tank wall to the pressure gauge). All pressures are extracted from MDSPLUS tree.

b) *.abi input file. This option means that neutral gas pressures will be loaded from *.abi input file, which name can be chosen in the text field on the right side.

c) Skip. This option means that currently loaded/displayed neutral gas pressures will be used. No load procedure for this category will be performed.

3.5.1.11 3D Mesh/Grid for calculation

These input profiles are discussed in chapter 3.3.2 and Appendix A.2.

There are three load options:

a) Beam/plasma geom. This option means that some optimal grid/mesh parameters will be calculated based on the current beam and plasma geometry. This default mesh is sparse and should be used for a fast trial run. For more accurate calculation the finer mesh is needed. For this the z_steps are recommended to be changed to 10 times smaller values.

b) *.abi input file. This option means that code grid/mesh parameters will be loaded from *.abi input file, which name can be chosen in the text field on the right side.

c) Skip. This option means that currently loaded/displayed code grid/mesh parameters will be used. No load procedure for this category will be performed.

The checkbox “Load all data from” is introduced to simultaneously control how all parameters are loaded. There are three options:

Default: This option allows return all the “Load” settings to some initial (default) values. These values are preset for the all used method of loading parameters for DNBI_ALCATOR beam.

“*.abi input file”: This option when choose sets up all the lower fields to load the parameters/data from “*.abi” file. By default the name consists of the “beam”+.abi” (DNBI_ALCATOR.abi)

Skip: This option means that currently loaded/displayed parameters/data will be used. No load procedure for this category will be performed.

In order to save the new Load settings (if they were changed) the “Save” button should be pressed. The window is closed by the “Close” button. Any unsaved changes are discarded.

When the correct load settings are prepared, the “LOAD” button should be pressed. The Session Log window will display more info on progress of “Load” routine.

After the “LOAD” routine is finished the area “LOAD input data needed for calculation” becomes insensitive and area “CONSTRUCT data arrays needed for the calculation” becomes sensitive (see Figure 3.5.4).

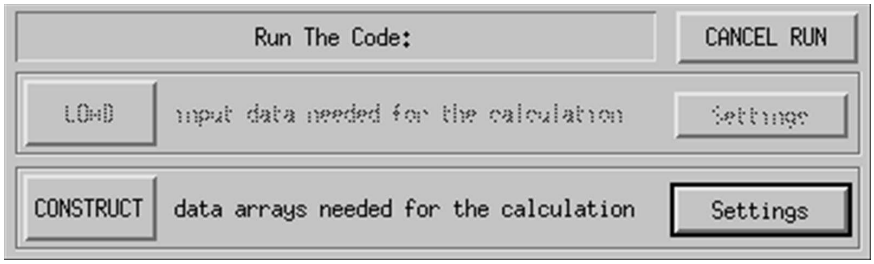


Figure 3.5.4 Construct data arrays needed for the calculation

At this point user has an option to save all the current “input parameters/data” into the “*.abi input file” (see Figure 3.5.5). This might be useful if user wants to be able to use these parameters for later runs. This *.abi input_file can be easily modified by any ASCII text editing software. A detailed description of the *.abi input_file is given in appendix C. A separate window is opened when user presses the “Save input data to (*.abi) input file” button. A user can enter the name of the file to which the input data to be saved. If the file with the same name exists, the data will be overwritten.

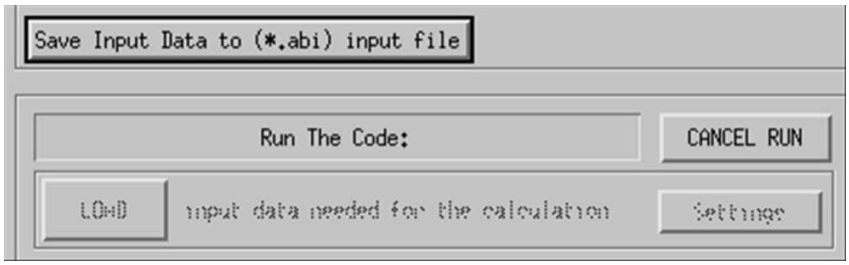


Figure 3.5.5 Save input data to (*.abi) input file

Note: After all the parameters are loaded, the routine builds four 1D arrays: **z_beam**, **x_beam**, **y_beam**, **e_beam** based on the “Code Grid/Mesh” and “Beam energy fractions” parameters. These arrays serve as a starting point for the 3D or 4D arrayswhich will be constructed (chapter 3.5.2)

3.5.2 Construct data arrays needed for the calculation

The “CONSTRUCT” button initiates a routine which constructs 3D arrays (calculated for each point of the Code grid) of input parameters needed for beam calculation. Before that the “Settings” button

should be pressed, which opens an additional window where user can select the methods by which these arrays should be constructed. A snapshot of this window is shown in Figure 3.5.6. Let's discuss each of these settings one by one:

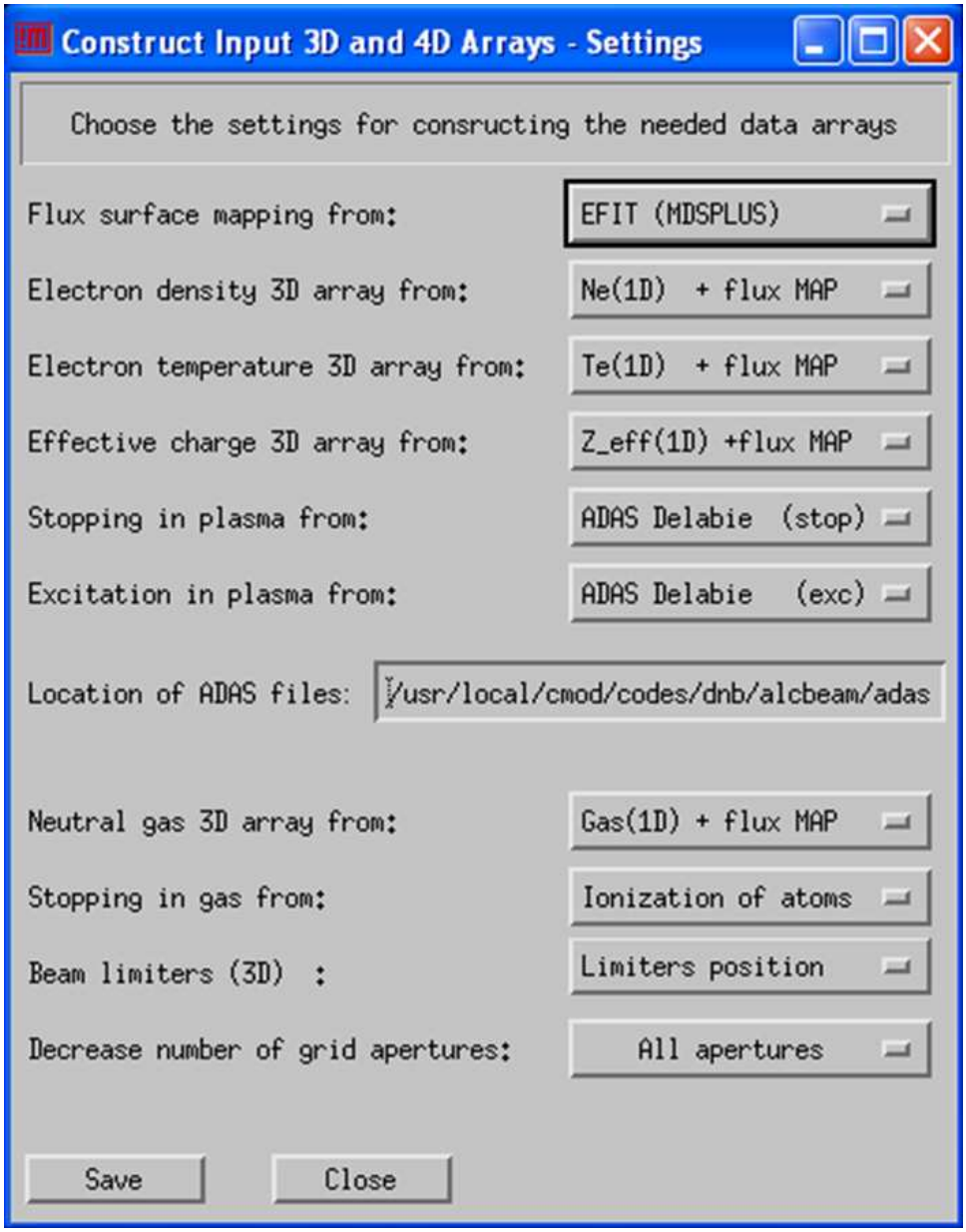


Figure 3.5.6 Construct input 3D and 4D arrays - Settings

3.5.2.1 Flux surface mapping

The location of the tokamak magnetic flux surfaces in reference to the beam position is needed to calculate a plasma parameter at any point in 3D space just based on 1D loaded profile. The assumption here is that plasma parameters are constant on flux surfaces.

There are three options how the positions of flux surfaces are calculated:

a) Miller equilibrium: The whole 3D structure of flux surfaces is calculated based on 6 plasma geometry parameters (described in chapter 3.3.5 and Appendix A.5). Ref: W.M.Stacey. Physics of Plasmas 15, 122505 (2008)
b) Hakkarainen equilibrium: The whole 3D structure of flux surfaces is calculated based on 6 plasma parameters (described in chapter 3.3.5 and Appendix A.5). Ref: Hakkarainen, Phys. Fluids B 2(7) 1990
c) EFIT (MDSPLUS). The whole 3D structure of flux surfaces is loaded from MDSPLUS EFIT tree. This option is only available for DNBI-ALCATOR beam and if Plasma Geometry input parameters are loaded from EFIT (MDSPLUS) (see chapter 3.5.1.5)
d) Skip. This option means that a previously constructed flux surfaces array will be used. No CONSTRUCT procedure for flux surfaces will be performed. This option should be used very carefully.

3.5.2.2 Electron density 3D array

There are two options how the 3D **n_e_arr** array is constructed:

a) **n_e** (1D) + flux MAP: A 3D array of electron density (**n_e_arr**) is constructed by means of (**z_beam**, **x_beam**, **y_beam**) arrays, 1D **n_e** loaded profile and constructed 3D array of flux surfaces (rho value).
b) Skip. This option means that a previously constructed **n_e_arr** 3D array will be used. No CONSTRUCT procedure for **n_e_arr** will be performed. This option should be used very carefully.

3.5.2.3 Electron temperature 3D array

There are two options how 3D **t_e_arr** array is constructed:

a) **t_e** (1D) + flux MAP: A 3D array of electron temperature (**t_e_arr**) is constructed by means of (**z_beam**, **x_beam**, **y_beam**) arrays, 1D **t_e** loaded profile and constructed 3D array of flux surfaces (rho value).
b) Skip. This option means that a previously constructed **t_e_arr** 3D array will be used. No CONSTRUCT procedure for **t_e_arr** will be performed. This option should be used very carefully.

3.5.2.4 Effective charge 3D array

There are two options how the 3D **z_eff_arr** array is constructed:

a) **z_eff** (1D) + flux MAP: A 3D array of plasma effective charge (**z_eff_arr**) is constructed by means of (**z_beam**, **x_beam**, **y_beam**) arrays, 1D **z_eff** loaded profile and constructed 3D array of flux surfaces (rho value).
b) Skip. This option means that a previously constructed **z_eff_arr** 3D array will be used. No CONSTRUCT procedure for **z_eff_arr** will be performed. This option should be used very carefully.

3.5.2.5 Stopping in plasma

A 4D array of beam-plasma stopping cross section needs to be constructed in advance for the code calculation. The array is constructed using the pre-constructed 3D arrays of: **n_e_arr**, **t_e_arr**, **z_eff_arr**, plasma impurity content (see chapter 3.3.8) and a chosen method of energy-dependent stopping cross section calculation.

There are three options for how stopping cross section 4D array are to be constructed:

a) ADAS Delabie (stop): A 4D array of beam-plasma stopping cross sections is constructed based on **n_e_arr**, **t_e_arr**, **z_eff_arr** 3D arrays, plasma impurity content (see chapter 3.3.8) and tabulated data extracted from ADAS files provided by Ephrem Delabie. This is the improved data for ADAS 3.0 with corrected cross-section error. The path to the directory, where ADAS files are located, is displayed below. The user can modify it. The array dimensions are: **e_beam**, **z_beam, x_beam**, **y_beam**.

b) ADAS vX.X (stop): A 4D array of beam-plasma stopping cross sections is constructed based on **n_e_arr**, **t_e_arr**, **z_eff_arr** 3D arrays, plasma impurity content (see chapter 3.3.8) and tabulated data extracted from ADAS vX_X files. The path to the directory, where ADAS files are located, is displayed below. The user can modify it. The domain is: **e_beam**, **z_beam, x_beam**, **y_beam**.
c) Suzuki (stopping): A 4D array of beam-plasma stopping cross sections is constructed based on **n_e_arr**, **t_e_arr**, **z_eff_arr** 3D arrays and empirical fitting formulas from Suzuki, Plasma Physics and Controlled Fusion, 40(1988). The domain is: **e_beam**, **z_beam**, **x_beam**, **y_beam**.
d) Skip. This option means that a previously constructed beam-plasma stopping cross section 3D array will be used. No CONSTRUCT procedure for "beam-plasma stopping cross section" will be performed. This option should be used very carefully.

3.5.2.6 Excitation in plasma

Two 4D arrays of beam excitation fractions **exc_n2_frac** and **exc_n3_frac** (n=2, 3) need to be constructed for the code calculation. The arrays are constructed by means of pre-constructed 3D arrays of: **n_e_arr**, **t_e_arr**, **z_eff_arr**, plasma impurity content (see chapter 3.3.8) and chosen method of excitation calculation. Beam excitation is always a beam dependent process, so the array dimensions are: **e_beam**, **z_beam**, **x_beam**, **y_beam**.

There are three options how excitation fractions 4D arrays are constructed:

a) ADAS Delabie (exc): Two 4D arrays of beam excitation fractions (n=2,3) are constructed based on **n_e_arr**, **t_e_arr**, **z_eff_arr** 3D arrays, plasma impurity content (see chapter 3.3.8) and tabulated data from ADAS files provided by Ephrem Delabie. This is the improved data for ADAS3.0 with corrected cross-section error. The path to the directory, where ADAS files are located, is displayed below. User can modify it.
b) ADAS vX_X (exc): Two 4D arrays of beam excitation fractions (n=2,3) are constructed based on **n_e_arr**, **t_e_arr**, **z_eff_arr** 3D arrays, plasma impurity content (see chapter 3.3.8) and tabulated data extracted from ADAS vX_X files. The path to the directory, where ADAS files are located, is displayed below. User can modify it.

c) Ian Hutchinson: Two 4D array of beam excitation fractions (n=2,3) are constructed based on **n_e_arr**, **t_e_arr**, **z_eff_arr** 3D arrays and tabulated data calculated by Prof. Ian Hutchinson code (Fortran code, was adapted to IDL). (ref: PPCF 44 71-82 (2002)). The data is saved in one of the code procedures.

d) Skip. This option means that a previously constructed beam excitation fractions 3D array will be used. No CONSTRUCT procedure for "beam excitation" will be performed. This option should be used very carefully.

3.5.2.7 Neutral gas 3D array

There are two options for how 3D neutral gas density array is constructed:

a) Gas (1D) + flux MAP: A 3D array of neutral gas density is constructed by means of (**z_beam**, **x_beam**, **y_beam**) arrays, 1D two value linear interpolation between **tank_pressure** and **torus_pressure** and constructed 3D array of flux surfaces (rho value).

b) Skip. This option means that a previously constructed neutral gas 3D array will be used. No CONSTRUCT procedure for "neutral gas" will be performed. This option should be used very carefully.

3.5.2.8 Stopping in gas

A 4D array of beam-gas stopping cross section needs to be constructed in advance in order to facilitate the code calculation. The array is constructed by means of pre-constructed 3D Neutral Gas arrays and chosen method of beam-gas stopping cross section calculation. Beam stopping in gas is always a beam dependent process so the array dimensions are: **e_beam**, **z_beam**, **x_beam**, **y_beam**.

There are two options how beam-gas stopping cross section 4D array is constructed:

a) Ionization of atoms: A 4D array of beam-gas stopping cross sections is constructed based on the Neutral Gas 3D array and an energy-dependent beam ionization cross section. The ionization cross section data has been extracted from spectroscopy code by Bob Granetz.

b) Skip. This option means that a previously constructed beam-gas stopping cross section 4D array will be used. No CONSTRUCT procedure for “beam-gas stopping cross section” will be performed. This option should be used very carefully.

3.5.2.9 Beam limiters

Attenuation of the beam due to the interaction with the beam limiters is calculated similarly to the stopping of the beam in plasma and gas. A point in the space has a 100% stopping coefficient if it has a limiter in it and 0% stopping coefficient if no limiter is present.

There are two options how 3D limiters array is constructed:

a) Limiters position: A 3D array of limiters position is constructed by means of (**z_beam**, **x_beam**, **y_beam**) arrays and inclusion of all the beam limiters according to their description parameters (see chapter 3.3.7 and Appendix A.7)

b) Skip. This option means that a previously constructed limiters 3D array will be used. No CONSTRUCT procedure for “beam limiters” will be performed. This option should be used very carefully.

3.5.2.10 Decrease the number of grid apertures (selected apertures)

This option allows for faster code calculation, when “code calculation grid” can’t be changed, This option decreases the number of the apertures used for the calculation. The resulted solution is an approximation to the “all apertures” solution. Note: this option allows for a fast run to test the configuration settings prior to a full run. It is recommended to run the “all aperture case” for the most realistic simulation.

There are several options. The user can either use “all apertures” (default value), or every 2 aperture, every 4 aperture, 6 aperture, ... etc. This means that the calculation will be performed 2, 4, 6 times faster.

In order to save the new Construct settings (if they were changed) the “Save” button should be pressed. The window is closed by the “Close” button. Any unsaved changes are discarded.

When the correct construct settings are prepared, the “CONSTRUCT” button should be pressed. The Session Log window will display more info on progress of “CONSTRUCT” routine.

So after the “CONSTRUCT” routine is finished the area “CONSTRUCT data arrays needed for the calculation” becomes insensitive and area “CALC beam code” becomes sensitive (see Figure 3.5.7)

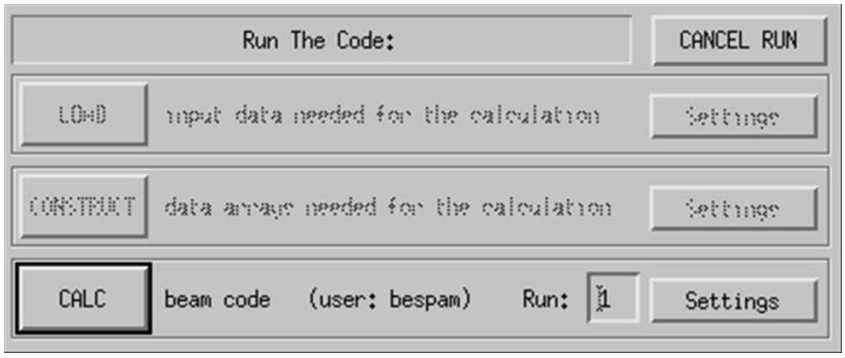


Figure 3.5.7 Calculation are of the main widget

3.5.3 Preview Constructed Arrays

After the “Construct” routine is completed, the “Preview constructed arrays” area also becomes sensitive (Figure 3.5.8). Similarly to way described in chapter 3.4 the constructed arrays can be previewed. Following is a description of constructed 3D and 4D array which can be previewed.

Used grid apertures: this option allows to plot (similar to the way described in chapter 3.3.10.1) the “selected aperture” (chapter 3.5.2.10)

n_e_arr: 3D array of the electron density (chapter 3.5.2.2) (**z_beam**, **r_major, x_beam**, **y_beam**)

t_e_arr: 3D array of the electron temperature (chapter 3.5.2.3) (**z_beam**, **r_major**, **x_beam**, **y_beam**)

z_eff_arr: 3D array of the plasma z_eff (chapter 3.5.2.4) (**z_beam**, **x_beam**, **y_beam**)

n0_arr: 3D array of the neutral gas (chapter 3.5.2.7) (**z_beam**, **x_beam**, **y_beam**)

exc_n2_frac (“n=2 exc”): 4D array of fraction on the atoms in the first excited state (n=2/n=1,2,3). (chapter 3.5.2.6) (**e_beam**, **z_beam**, **r_major**, **x_beam**, **y_beam**)

exc_n3_frac (“n=3 exc”): 4D array of fraction on the atoms in the second excited state (n=2/n=1,2,3). (chapter 3.5.2.6 (**e_beam**, **z_beam**, **r_major**, **x_beam**, **y_beam**)

limiters: 3D array of the volume occupied by limiters (chapter 3.5.2.9) (**z_beam**, **x_beam**, **y_beam**)

Figure 3.5.8 Preview constructed arrays

3.5.4 Calculation of the beam divergence and penetration

The final calculation/simulation can be initiated by “CALC” button. Before pressing "CALC", the Run number should be chosen and “Settings” button should be pressed. The Run number will be used to select the MDSPLUS node or compose a name of the **output_file** where calculation results are saved. If a MDSPLUS node or **output_file** with the same number is previously existed, the data will be overwritten.

The “Settings” button opens an additional window where user can select the simulation methods and optional flags for how the calculation is performed. Also the user can select where the output data is to be saved. A snapshot of this window is shown in Figure 3.5.9. Let’s discuss each of these settings one by one.

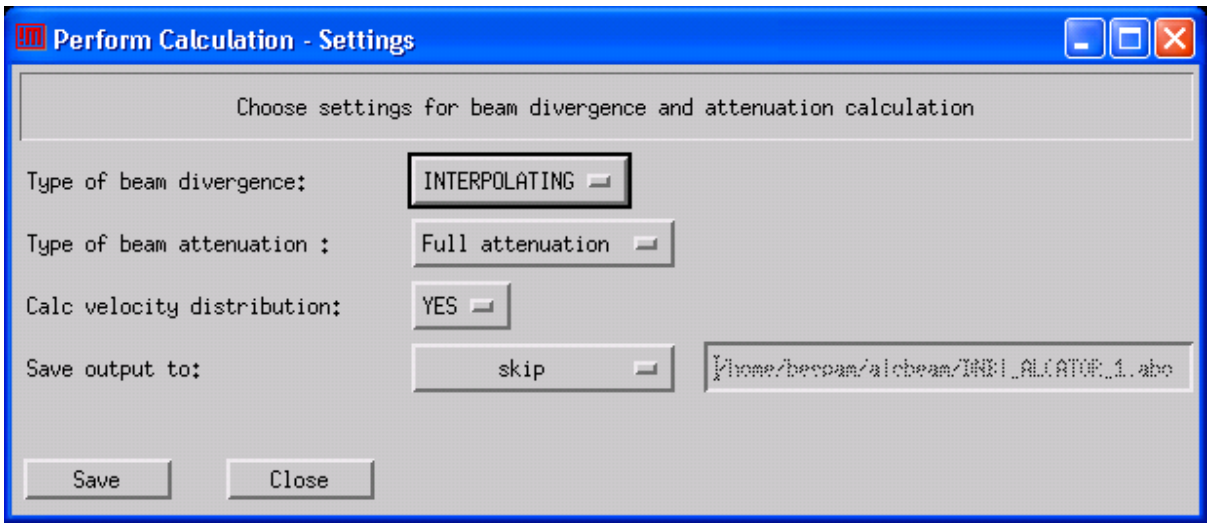


Figure 3.5.9 Perform calculations - Settings

3.5.4.1 Type of beam divergence

There are three available methods for divergence and penetration of the beam to be calculated.

a) RAY TRACING method: The beam divergence is simulated as a sum of elemental beamlets extracted from each aperture, each of which is diverged with a defined divergence angle. A straight line attenuation model is used on every ray connecting a grid

aperture with a node on the mesh. This trajectory is divided into many steps, in which the attenuation is calculated from the pre-calculated 4D local attenuation cross-section array. A detailed description of this method is given in Appendix H. This type of calculation has the highest accuracy, but is the most time-consuming.

b) INTERPOLATION method: This method is similar to RAY TRACING. The only difference is that the beam attenuation, similar to attenuation cross-section, is calculated beforehand by interpolation method, which greatly improves the code performance. Error is introduced due to the increased error in attenuation factors. A detailed description of this method is given in Appendix I. This type of calculation is optimal for most of the cases (average performance, high accuracy). This method should not be used for cases when beam is attenuated strongly by limiters.

c) "ANALYTIC" method: The main advantage is a fast calculation. This method is not as flexible as other methods. The solution is less accurate, since it is an approximate solution. This method also cannot calculate the beam attenuation and shouldn't be used if beam attenuation is needed. A detailed description of this method is given in Appendix J.

3.5.4.2 Type of beam attenuation

User has an option to turn ON/OFF some of the attenuation processes for simulation. This feature has proved to be useful in the testing phase.

There are five options here:

a) Full attenuation: all attenuation methods are included in calculation/simulation

b) Plasma only: only beam attenuation in plasma is included in calculation. Beam attenuation processes in neutral gas and by limiters are omitted.

c) Gas only: only beam attenuation in neutral gas is included in calculation. Beam attenuation processes in plasma and by limiters are omitted.

d) Limiters only: only beam attenuation by limiter structures is included in calculation. Beam attenuation processes in plasma and in neutral gas are omitted.

e) Gas + limiters: beam attenuation by neutral gas and limiter structures is included in calculation. Beam attenuation by plasma is omitted.

f) Skip attenuation: No attenuation is taking into the account.

If “Analytic” beam divergence is chosen, the type of beam attenuation droplist is set to “Skip attenuation” value and is insensitive.

3.5.4.3 Calc velocity distribution

YES: In addition to beam density calculation calculate velocity distribution (takes longer)

NO: Calculate only beam densities.

3.5.4.4 Save output to:

The user has an option to save the output result either to an MDSPLUS node (for DNBI_ALCATOR beam only), save to an *.abo output file or not to save the results at all. The format of the output data is described in Appendix B.

Detailed descriptions of each of the options are:

a) MDSPLUS: The output data (also used input parameters/data) is saved into the ALCATOR MDSPLUS node. The name of the node composed from the “Username” and “Run” number (See chapter 3.5) (example: DNB.ALCBEAM.BESPAM.RUN_1). The full description of the format of the data saved into the MDSPLUS node is given in Appendix C.
b) *.abo output file: The output data (also used input parameters/data) is saved to the “*.abo output_file”. The name of the **output_file** can be chosen in the text filed on the right. The full description of the data format saved into the *.abo **output_file** is given in appendix D.
c) Skip: The output data is temporary saved to the code environment (so it can be previewed), but not to be saved to MDSPLUS node or **output_file**.

In order to save the new Calc settings (if they were changed) the “Save” button should be pressed. The window is closed by the “Close” button. Any unsaved parameters are discarded.

When the correct calc settings are prepared, the “CALC” button should be pressed. The Session Log window will display more info on progress of “CALC” routine.

After the “CALC” routine is started, the area “Calc beam code” becomes insensitive.

An estimated time counter will be also shown. The user has an option to STOP the calculation by pressing the “Stop Calc” button (see Figure 3.5.10.)

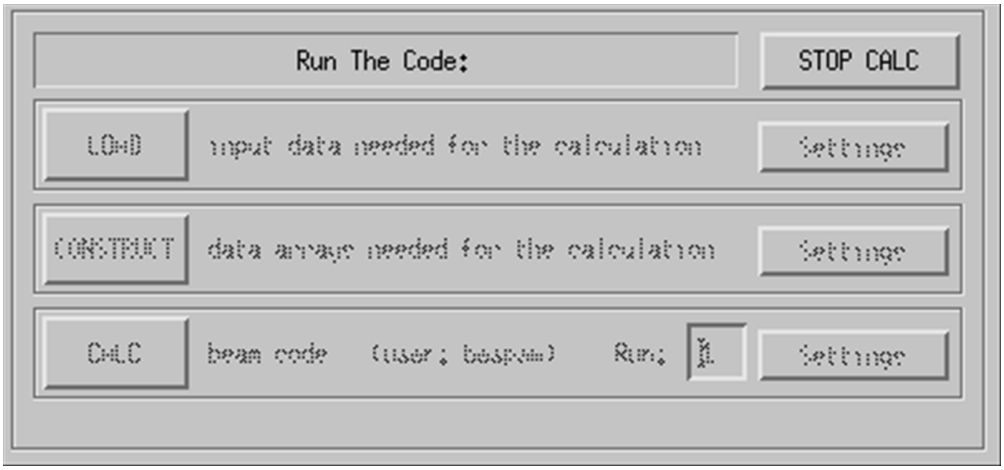


Figure 3.5.10 Stop Calc

If calculation is stopped the “Calc beam code” area again becomes sensitive if user wants to rerun the code with different “Calc” settings. (see Figure 3.5.11). The code can be returned to “preview state” by pressing the “CANCEL RUN” button. No output data will be available in this case.

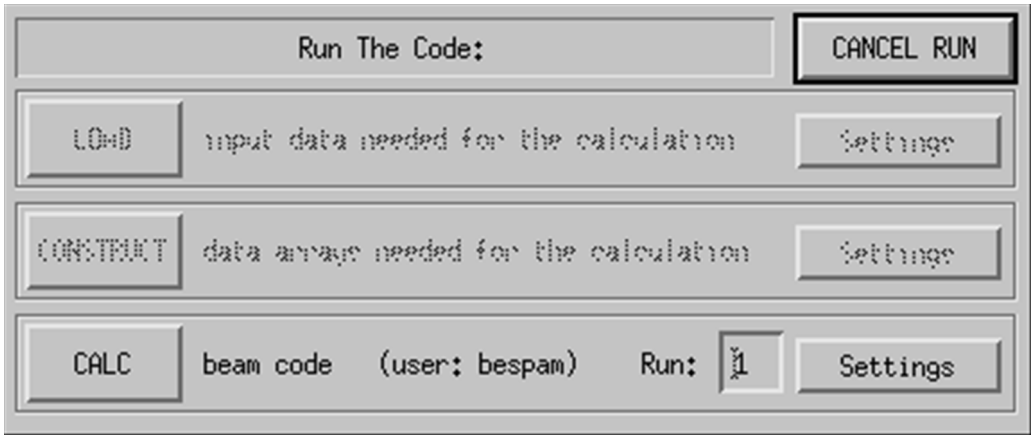


Figure 3.5.11 Cancel run

If calculation was finished the “Calc beam code” area again becomes sensitive if user wants to rerun the code with different “Calc” settings. (see Figure 3.5.12). The code can be returned to “preview state” by pressing the “FINISH RUN” button.

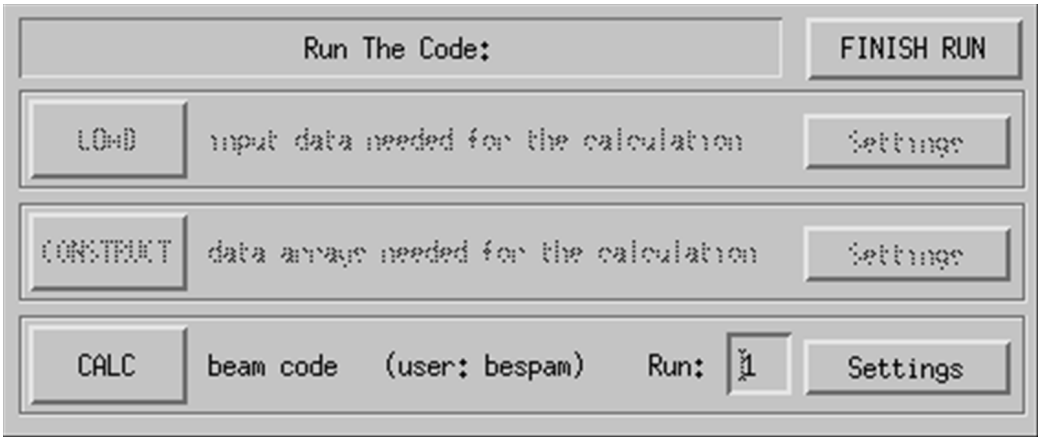


Figure 3.5.12 Finish run

3.6 Save/load setting to/from (*.aset) file

There are two buttons at the bottom of the “Run the Code” field (see Figure 3.5.1). These two buttons activate additional GUI windows (see Figure 3.6.1 and Figure 3.6.2) , for user to save/load the current (LOAD/CONSTRUCT/CALC) setting from/to a single *.aset file. This file can be used to save the settings for the later runs or by ALCBEAM driver (see chapter 3.7). The format of the *.aset file is discussed in the Appendix E.

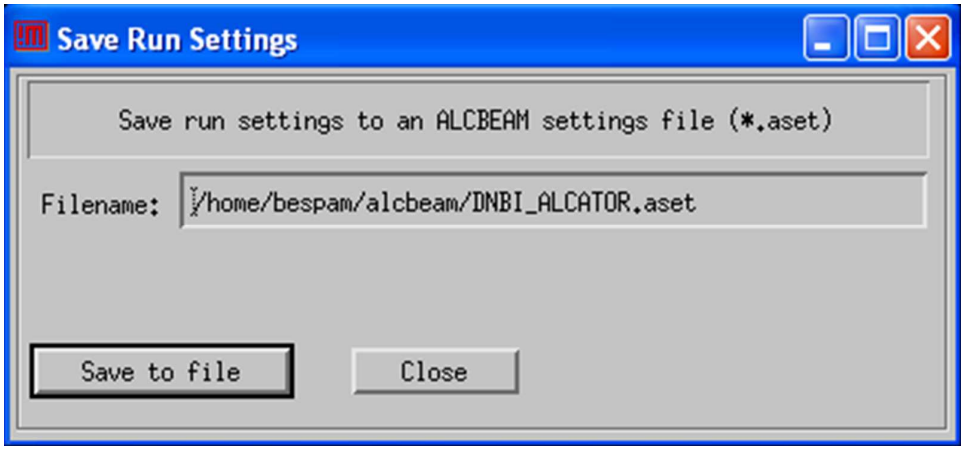


Figure 3.6.1 Save Run settings

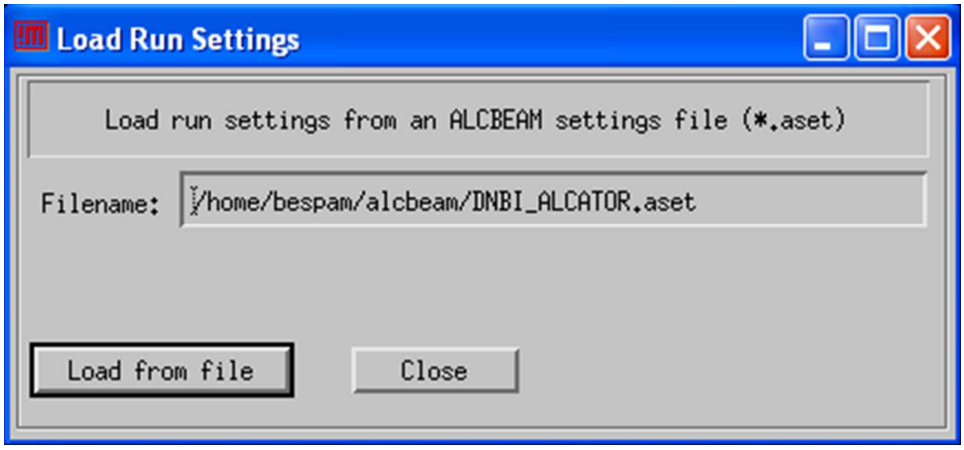


Figure 3.6.2 Load Run settings

3.7 ALCBEAM driver.

The "Prepare ALCBEAM Driver" button opens the driver window (see Figure 3.7.1) and puts the widget in ALCBEAM driver operation mode. Here the user can select the (*.aset) settings file (see chapter 3.6) which will be used for following ALCBEAM runs. Then range of shots or list of shots separated by the comma should be selected. “Run Driver” button initiates the calculation. Once the driver has started the “Run Driver” button becomes the “Pause Driver” button, which will pause the calculation on user request. If driver calculation was paused it can be continued by pressing “Continue” button. The “Quit Driver” button will close the “driver window and return to normal ALCBEAM operation mode.

Currently driver mode can only be used for C-Mod’s DNBI simulations. The driver finds optimal simulation time intervals, according to the DNB current modulation time trace, and run ALCBEAM for these intervals. The rest of the input parameters are imported according to the methods specified in the settings file. Data is saved also according to the method specified in the settings file. The name modifier for the first ALCBEAM RUN is set up to be 1. The following ALCBEAM runs will have the name modifiers incrementally increased: 1,2,3,4,5,..N. Where N is the number of ALCBEAM simulations run by driver for a certain shot number.

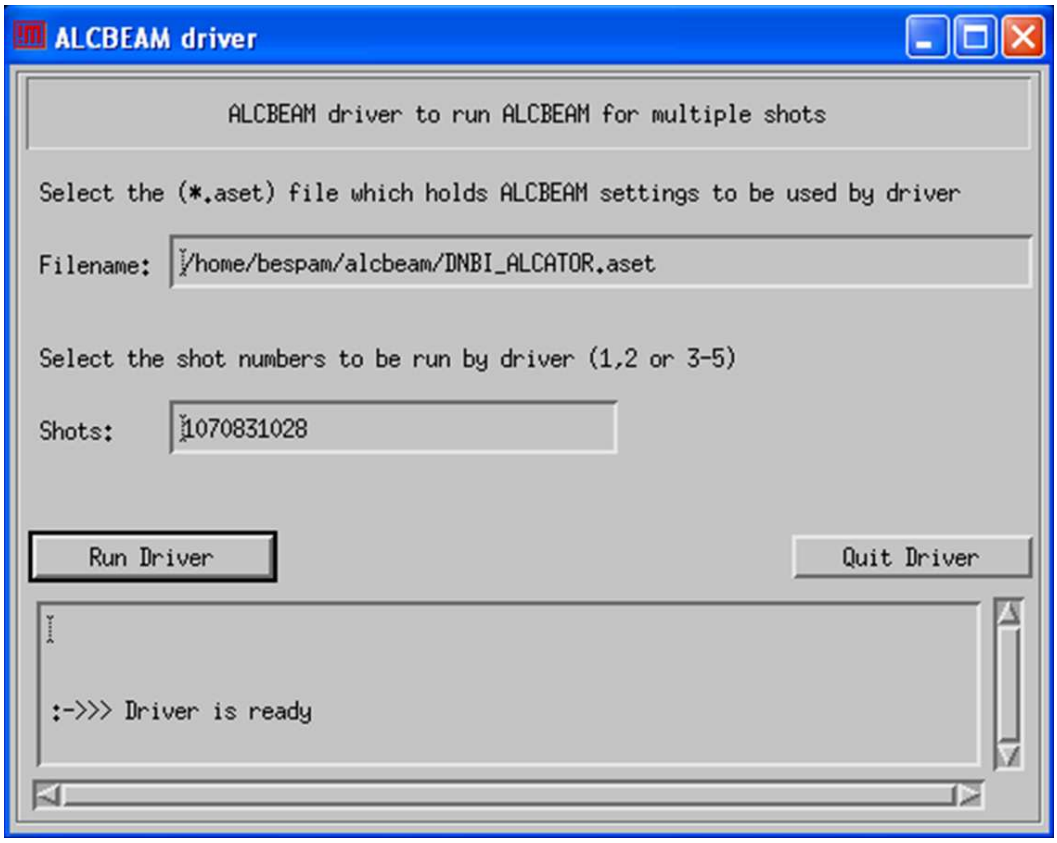


Figure 3.7.1 ALCBEAM driver window

3.8 Export plots

Any of the graphs/data plotted by ALCBEAM can be exported as *.png image or ascii-text (*.adat) file. The “Export” button (see Figure 3.4.1) opens additional settings window (see Figure 3.8.1), where use can select the filename and select the type of export. The format of the *.aset file is discussed in the Appendix F.

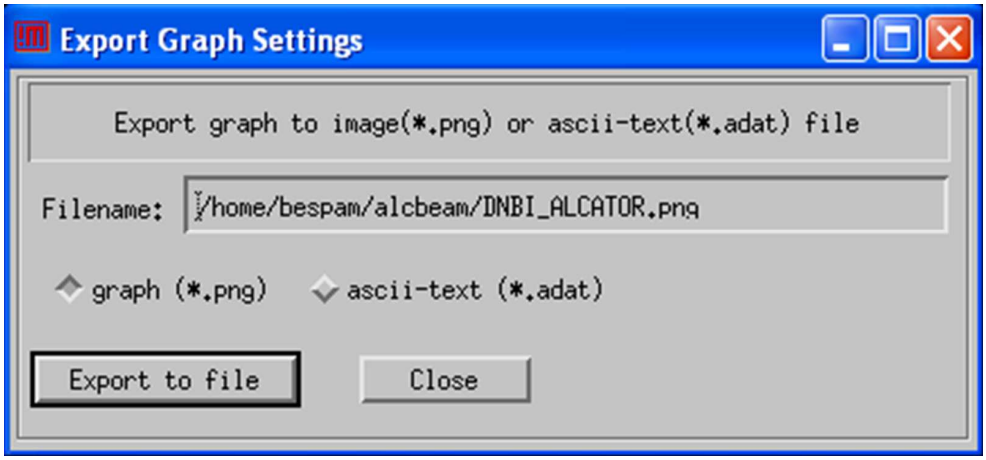


Figure 3.8.1 Export Graph Settings

Appendix A Input parameters and data.

The beam simulation code requires some input parameters and data for calculation. A detailed description of these parameters is given below.

The parameters/data are divided in 9 categories (see chapter 3.3)

A.1. General parameters (see chapter 3.3.1):

beam ("Beam Type") : This literal label is used to distinguish the simulated beams. This label is frequently used by the code for default names of the input and output files. User is free to use any beam label. However it is recommended to use the following pattern (example: DNBI_ALCATOR, means (type of the beam injector (DNBI) @ type of the tokamak (ALCATOR)). The label DNBI_ALCBEAM is unique and must be used for ALCATOR simulations, since user gets access to the ALCATOR MDSPLUS database only if beam is DNBI_ALCATOR.

shot ("Selected Shot"): This numerical label identifies the plasma shot for which simulation is made. User free to use any shot label. However it is recommended to use the real tokamak shot number in case if it is appropriate. In case of DNBI_ALCBEAM beam the shot label identifies the MDSPLUS tree from which the input data can be extracted and output data can be saved.

time_interv ("Selected time interval"). Array of two parameters (dimension, sec) identify the time interval of the shot for which the simulation is/was made. As it was discussed in chapter 1, ALCBEAM code simulates the beam densities at a particular time. All non-stationary input parameters will be averaged along this time interval and the average value used by the code.

A.2. Code Grid/Mesh (see chapter 3.3.2):

code_grid_array : There are 11 geometrical (dimension, m) parameters, which define how fine/coarse the 3D geometrical calculation grid/mesh will be. By means of these parameters the arrays (**z_beam**, **x_beam**, **y_beam**) are constructed. **x_beam** and **y_beam** identify two dimensions which form the plane perpendicular to direction of the beam propagation **z_beam**. **Y_beam** is the direction parallel to the Z-axis of the Tokamak (bottom->top). **x_beam** axis is defined as a cross product of **z_beam** and **y_beam**. There are 11 parameters:

(X_beam_Min, X_beam_Step, X_beam_Max): **x_beam** is constructed as a regular X-coordinate array (started from X_beam_Min, ended at X_beam_Max, with step of X_beam_Step). Note. If X_beam_Step is not a divisor of (X_beam_Max-X_beam_Min) then the (X_beam_Max-X_beam_Min) interval is divided in several intervals with step close to X_beam_Step.

(Y_beam_Min, Y_beam_Step, Y_beam_Max): **y_beam** is constructed as a regular Y-coordinate array (started from Y_beam_Min, ended at Y_beam_Max, with step of Y_beam_Step). Note. If Y_beam_Step is not a divisor of (Y_beam_Max-Y_beam_Min) then the (Y_beam_Max-Y_beam_Min) interval is divided in several intervals with step close to Y_beam_Step.

(Z_beam_Min, Z_beam_Step1, Z_beam_Mid, Z_beam_Step2, Z_beam_Max): **z_beam** is constructed as a compound of two regular Z-coordinate arrays (first started from Z_beam_Min, ended at Z_beam_Mid, with step of Z_beam_Step1, second started from Z_beam_Mid, ended at Z_beam_Max, with step of Z_beam_Step2). Note. Similar procedure of regulation (discussed above) is applied if Z_beam_Step1 and Z_beam_Step2 are not divisors of (Z_beam_Mid-Z_beam_Min) and (Z_beam_Max-Z_beam_Mid) correspondingly.

All these parameters are saved in the code environment as a structure:

code_grid_arr={z:[Min,Step1,Mid,Step2,Max],x:[Min,Step,Max],y:[Min,Step,Max]

A.3. Beam Geometry (see chapter 3.3.3):

grid_ap_diam: Diameter of elemental aperture in the extraction grid (units, mm). Currently not used in the simulation, but used in some of the figures.

beam_port: Tokamak horizontal port (typically a literal or numerical value, example “F”) to which the beam injector is connected.

x_grid_focus: Radius of curvature of the accelerating grids along the X direction (units, m).

y_grid_focus: Radius of curvature of the accelerating grids along the Y direction (units, m).

x_bml, y_bml: These are the two array of geometrical X and Y coordinates of location of apertures in beam accelerating grid.

grid_ap_diam: Diameter of elemental aperture in extraction grid (units, m).

tank_front_dist: Distance (along z_beam direction) from center of the accelerating grid to the tank closest/front wall (units, m).

tank_size: Length (along z_beam direction) of the beam vacuum tank (units, m).

neutr_size: Length (along z_beam direction) of the neutralizer tube (units, m). The neutralizer starts from the accelerating grids and extends into the beam vacuum tank.

tank_magnet_dist: Distance (along z_beam direction) from front wall of the beam vacuum tank to the front/closest side of the “ions deflection” magnet (units, m).

magnet_size: Length (along z_beam direction) of the ions deflection magnet (units, m).

tank_cal_dist: Distance (along z_beam direction) from second/back wall of the beam vacuum tank to the center of the calorimeter (units, m). Currently not used in the simulation, but used in some of the figures.

Two point which effectively defines the orientation of the beam relative to tokamak

r_grid: Major radius (in tokamak coordinates) of the center of the accelerating grid (beam origin).

z_grid: Coordinate Z (in tokamak coordinates) of the center of the accelerating grid (beam origin).

phi_grid: Coordinate phi (in tokamak coordinates) of the center of the accelerating grid (beam origin).

r_wall: Major radius (in tokamak coordinates) of the intersection of beam centerline with the tokamak wall.

z_wall: Coordinate Z (in tokamak coordinates) of the intersection of beam centerline with the tokamak wall.

phi_wall: Coordinate phi (in tokamak coordinates) of the intersection of beam centerline with the tokamak wall.

A sketch of geometrical locations of all the beam injector parts can be previewed in the "Beam Geometry" window (see chapter 3.3.3).

A.4. Neutral Gas (see chapter 3.3.4):

tank_pressure: Neutral Gas pressure (averaged over [t1, t2]) in the beam vacuum tank (units, mtorr).

torus_pressure: Neutral gas pressure (averaged over [t1, t2]) in the tokamak vessel (units, mtorr).

duct_pressure: Neutral gas pressure (averaged over [t1, t2]) in the beam duct (units, mtorr).

duct_pressure_loc: Distance from the nearest tank wall to the location of the duct pressure measurement (units, m).

These pressures are used to calculate the distribution of the neutral gas on the way of the beam. This is used to calculate the beam attenuation in the neutral gas.

A.5. Plasma Geometry (see chapter 3.3.5):

r_major: Major radius of the plasma in the tokamak. Radial coordinate (R_tokamak) of the center of the plasma column (units, m).

z_major: Vertical position (Z_tokamak) of the plasma center (units, m).

r_minor: Minor radius of the plasma in the tokamak. Radius of the plasma column (units, m).

elong: Elongation of the plasma in the tokamak (non-dimensional).

triang_upper: Upper triangularity of the plasma in the tokamak (non-dimensional).

triang_lower: Lower triangularity of the plasma in the tokamak (non-dimensional).

These parameters are enough to define the 3D positions of the plasma magnetic flux surfaces.

A.6. Beam Parameters (see chapter 3.3.6 and 3.3.10):

beam_atom: (H, D or T) depending on the beam fueling gas.

e_full: Energy of the main/full energy component (units, keV).

e_frac: Beam energy components (fractions of the e_full) (non-dimensional). Each beam has a few groups of atoms with different energy components. Typically each group has an energy which is a fraction of the **e_full**. (Example: [1.0000, 0.500, 0.330, 0.055], which means that beam has [full, 1/2, 1/3 and 1/18] energy components.

i_beam: Total extracted current of the beam (units, A).

i_frac: Fractions of the full beam extracted current associated with each beam energy component.

i_dens_par: β parameter (Eq. 1), which is used to quantify the distribution of the extracted current among all of the apertures in the accelerating grid. The distribution has parabolic shape according to Eq. 1.

$$I_{bml} = I_{bml}^{cent} \times \left[1 - \beta \times \left(x_{bml}^2 + y_{bml}^2\right)\right], \sum_{\substack{all \\ pores}} I_{bml} = I_0 \quad (1)$$

Extraction current is distributed uniformly among the apertures if **i_dens_par** (β)=0.

i_opt: Optimal extraction current, when beam divergence is minimal. Optimal extraction current is usually measured empirically by searching for current when beam has minimal size at calorimeter.

x_div_bml_opt and **y_div_bml_opt.**: Optimal divergences of the elemental beamlet of the beam in X and Y directions (units, deg). A beamlet is an elemental part of the beam extracted from a single aperture. Typically the beamlet is propagated toward the beam focal point defined by (**x_grid_focus** and **y_grid_focus**) and has a Gaussian distribution with 1/e x_width proportional to tan(**x_div_bml**), 1/e y_width proportional to tan(**y_div_bml**) and a distance from the aperture. It is assumed that beamlet has an optimal divergence when extracted beamlet current (I_{bml}) is optimal. Beamlet divergence changes according to the Eq. 2.

$$\theta(I_{bml}) \approx \theta_{\min}\left(1+\gamma(1-I_{bml}/I_{bml}^{opt})^2\right) \quad (2)$$

div_dist_par: parameter γ in Eq 2.

These parameters are all other non-geometrical beam parameters, which are needed for accurate beam simulation.

A.7. Beam Limiters (see chapter 3.3.7):

n_limiters: Number of elements, which constitute the limiter/aperture structure for the beam (non-dimensional). This number defines the number of columns in the **limiters_table** array.

limiters_table: A 2D array (1D if only n_limiters=1)of values, which describe the geometrical positions of all the beam limiters/apertures. There are 6 rows in the limiters_table array: [Name, Z_pos, Z_size, Diameter, X_size, Y_size]. Z_pos is the z_beam coordinate of the front (closest to the accelerating grid) side of the limiter.

There are two types of limiters/apertures user can specify:

Cylindrical limiters/apertures (example: beam duct, aperture, etc) are defined by (Z_pos, Z_size, Diameter) values. Note: The values (X_size, Y_size) should be setup to be 'NAN' for cylindrical limiters/apertures.

Rectangular limiters/apertures (example: gate, aperture, etc) are defined by (Z_pos, Z_size, X_size, Y_size) values. Note: The value (Diameter) should be setup to be 'NAN' for rectangular limiters/apertures.

Tokamak walls (examples: inner or outer walls) are defined by a name and one geometrical parameter: (R_major). Other parameters should be setup to be 'NAN'.

A.8. Plasma parameters (see chapter 3.3.8):

Some global plasma parameters, which is needed for beam attenuation. In order to calculate the correct beam stopping cross-section (and excitation fractions) from ADAS files the relative impurity content needs to be known. ADAS stopping cross-sections and relative excited fractions are calculated for different impurities and saves as separate files. For ALCBEAM v 2.9 impurity files for B, Ar and C are the only available. So if ADAS source is chosen, the impurities should be selected are only B, Ar and C. Other impurities could be added later on request. Z_eff profiles define the total impurity content in the plasma.

There are 3 different plasma parameters in this category:

main_ion: By default it is D. Supported values are: H, D, He.

n_impur: Number of impurity species.

impur_table: table of (3x**n_impur**). First row: impurity_label. Second row: Impurity ion charge (5,18,6). Third row: Fractions. Total should be 1.0.

A.9. Electron density (n_e), temperature (t_e) and z_eff profiles (see chapter 3.3.9):

Electron density (**n_e**): 1D profiles of the electron density and its error. The profiles measured/mapped to the tokamak midplane, which (preferable) covers the full spatial range of the outer half of the plasma column.

n_e_raw_r: Radial positions (major radius) of the electron density profile (units, m).

n_e_raw: Measured electron density profiles at **n_e_raw_r** (units, cm^{-3}).

n_e_raw_err: Measured electron density error for **n_e_raw** values (units, cm^{-3}).

n_e_r: Radial positions (major radius) of the smoothed electron density profile (units, m).

n_e: Smoothed electron density profiles at **n_e_raw_r** (units, cm^{-3}). It is calculated based on the **n_e_raw** and smoothed by the smoothing method discussed in chapter 3.3.9. This array can also be loaded directly from the *.abi input_file or *.abo output_file (see see Appendix C and Appendix D).

n_e_err: Smoothed electron density error for **n_e_raw** values (units, cm^{-3}). It is interpolated from **n_e_raw_err** and **n_e_raw_r** onto the interval **n_e_r.** This array can also be loaded directly from the *.abi input_file or *.abo output_file (see Appendix C and Appendix D).

Electron temperature (**t_e**): 1D profiles of the electron temperature and its error. The profiles measured/mapped to the tokamak midplane, which (preferable) covers the full spatial range of the outer half of the plasma column.

t_e_raw_r: Radial positions (major radius) of the electron temperature profile (units, m).

t_e_raw: Measured electron temperature profiles at **t_e_raw_r** (units, keV).

t_e_raw_err: Measured electron temperature error for **t_e_raw** values (units, keV).

t_e_r: Radial positions (major radius) of the smoothed electron temperature profile (units, m).

t_e: Smoothed electron temperature profiles at **t_e_raw_r** (units, keV). It is calculated based on the **t_e_raw** and smoothed by the smoothing method discussed in chapter 3.3.9. This array can also be loaded directly from the *.abi input_file or *.abo output_file (see see Appendix C and Appendix D).

t_e_err: Smoothed electron temperature error for **t_e_raw** values (units, keV). It is interpolated from **t_e_raw_err** and **t_e_raw_r** onto the interval **t_e_r**. This array can also be loaded directly from the *.abi input_file or *.abo output_file (see see Appendix C and Appendix D).

Effective charge (**z_eff**): 1D profiles of the z_eff and its error. The profiles measured/mapped to the tokamak midplane, which (preferable) covers the full spatial range of the outer half of the plasma column.

z_e_raw_r: Radial positions (major radius) of the **z_eff** profile (units, m).

z_e_raw: Measured z_eff profiles at **z_eff_raw_r** (units, keV).

z_e_raw_err: Measured z_eff error for **z_eff_raw** values (units, keV).

z_e_r: Radial positions (major radius) of the smoothed z_eff profile (units, m).

z_e: Smoothed z_eff profiles at **z_eff_raw_r** (units, keV). It is calculated based on the **z_eff_raw** and smoothed by the smoothing method discussed in chapter 3.3.9. This array can also be loaded directly from the *.abi input_file or *.abo output_file (see Appendix C and Appendix D).

Appendix B Output results/data.

Here the format of the output data is discussed.

e_beam: The array of the energies of each component

z_beam: The array of the output z coordinated

x_beam: The array of the output x coordinated

y_beam: The array of the output y coordinated

n_beam: Is the main 4D output array of beam density (**e_beam**, **z_beam**, **x_beam**, **y_beam**). This array holds the value of density of atoms in aground and excited states. Only two first excitation states of the atoms are taking into the account here.

So in order to extract the density of the atoms in the first excited state we need to multiply **n_beam** by **exc_n2_frac** (see chapter 3.5.2.6). In order to extract the density of the atoms in the second excited state we need to multiply **n_beam** by **exc_n3_frac** (see chapter 3.5.2.6).

Similarly (**n_beam- n_beam* exc_n3_frac - n_beam* exc_n3_frac**) is the density of the atoms in the ground state.

If **calc velocity distribution** option (see chapter 3.5.4.4) is enabled then output file/node has some additional parameters:

vel_vec_x: 3D array (n_x, n_y, n_bml) of X Coordinates of the velocity vector distribution, m

vel_vec_y: 3D array (n_x, n_y, n_bml) of Y Coordinates of the velocity vector distribution, m

Note: z_beam is supposed to be used as remaining Z coordinate.

vel_vec_coef: 5D array (n_e, n_z, n_x, n_y, 9) of polynomial fit coefficients for XY velocity distribution.

Note: How to get a velocity distribution for each point on 3D beam mesh (n_x,n_y, n_z)

Distribution consists of contributions from each beamlet pore/aperture. If we decided to save directly each contribution from a pore we would end up have an array of size 1G for a standard ALCBEAM run (or 10GB for a fine mesh run). In order to make a data save possible, we fit a 2D velocity distribution by a polynomial surface (9 coefficients) $f(x,y)=[k(0), k(1)*y, k(2)*y^2, k(3)*x, k(4)*xy, k(5)*xy^2, k(6)*x^2, k(7)*x^2y, k(8)*x^2y^2]$.

How to extract a velocity distribution at a point (x,y,z) for beam energy component e_i

x_i=locate(x_beam,x)

y_i=locate(y_beam,y)

z_i=locate(z_beam,z)

Velocity vector from each beamlet pore: [vel_vec_x(x_i,y_i,n_bml), vel_vec_y(x_i,y_i,n_bml), z]

Set of 9 coefficients (k): vel_vect_coef (e_i, z_i, x_i, y_i, 9)

Amplitude for each vector: f(vel_vec_x(x_i,y_i,n_bml), vel_vec_y(x_i,y_i,n_bml))

Sum of all this vectors gives a full distribution.

Appendix C Format of the *.abi input file.

The *.abi file holds all input parameters/data necessary to run the ALCBEAM simulation. Here the format of this file is discussed.

An example of a DNBI_ALCATOR.abi file is shown below. The file structure is preserved. This manual shows additional explanatory comments (beginning with * and highlighted in blue), which should be removed from the real file.

;Input parameters for ALCBEAM code

;File is created on:
Thu Sep 3 10:22:43 2009
;File is created by:
ALCBEAM (ver. 2.9)

;Beam label
beam:
DNBI_ALCATOR

;Shot number
shot:
1070831028

;Time interval t1, t2
time_interv:
1.080, 1.100

*---------- this part of the file is similar for *.abi file and *.abo file----------

---

;X positions of the apertures in the accelerating grid, m
x_bml:
* Here should be x_bml values separated (by one comma and space)

;y positions of the apertures in the accelerating grid, m
y_bml:
* Here should be y_bml values separated (by one comma and space)

;Diameter of elemental aperture in extraction grid, mm
grid_ap_diam:

4.000

;Beam injector is attached to the port
beam_port:
F

;Radius of curvature of the grids (X:horizontal), m
x_grid_focus:
3.400

;Radius of curvature of the grids (Y:vertical), m
y_grid_focus:
3.400

;Distance from grids to the tank front wall, m
tank_front_dist:
0.500

;The size of the beam vacuum tank, m
tank_size:
1.650

;The size of the beam neutralizer tube, m
neutr_size:
1.000

;Distance from tank front wall to the front of magnet, m
tank_magnet_dist:
0.684

;The size of the beam deflection magnet, m
magnet_size:
0.191

;Distance from tank end wall to front of calorimeter, m
tank_cal_dist:
0.170

;Major radius of center of beam accelerating grid, m
r_grid:
5.398

;Z coordinate of center of beam accelerating grid, m
z_grid:
0.000

;Toroidal angle of center of beam accelerating grid, m
phi_grid:
-0.096

;Major radius of second point defining the beam, m
r_wall:
0.440

;Z coordinate of second point defining the beam, m
z_wall:
0.000

;Toroidal angle of second point defining the beam, m
phi_wall:
0.205

;Beam atom:
beam_atom:
H

;Energy of the main component, keV
e_full:
48.802

;Beam energy components (fractions)
e_frac:
1.000, 0.500, 0.333, 0.056

;Total current of the beam, A
i_beam:
5.454

;Current fraction of each component
i_frac:
0.803, 0.059, 0.108, 0.030

;Optimal extraction current, I_opt, A
i_opt:
5.600

;Parabolic source density parameter
i_dens_par:
0.000

;beamlet divergence half-angle (X:horizontal), deg
x_div_bml_opt:

0.700

;beamlet divergence half-angle (Y:vertical), deg
y_div_bml_opt:
0.700

;Beamlet divergence variation parameter
div_dist_par:
5.000

;Number of beam limiters to set up
n_limiters:
1

;Table of limiters positions and sizes
;If limiter is circular X size and Y size should be blank or 0.000, Diameter should be non zero
;If limiter is rectangular Diameter should be blank or 0.000, X size and Y size should be non zero
limiters_table:
---
name : Z pos : Z size :Diameter: X size:Y size: R major
---
beam_duct : 2.150: 2.250: 0.2: NAN: NAN: NAN

;Pressure in the beam tank, mtorr
tank_pressure:
0.300

;Pressure in the residual gas in the torus, mtorr
torus_pressure:
0.200

;Distance from the tank wall to the duct pressure gauge, m
duct_pressure_loc:
1.000

;Pressure in the duct, mtorr
duct_pressure:
0.250

;Major radius of the tokamak, radius of plasma center), m
r_major:
0.678

;Vertical position of the plasma center, m
z_major:

-0.011

;Minor radius of the tokamak, m
r_minor:
0.219

;Elongation of the plasma
elong:
1.641

;Upper triangularity of the plasma
triang_upper:
0.371

;Lower triangularity of the plasma
triang_lower:
0.562

;Main Plasma Ion
main_ion:
D

;Number of plasma impurities to set up
n_impur:
2

;Table of impurities (symbol, ion charge, and density fractions relative to total density of all impurities)
;At least one impurity column should be selected
impur_table:

Label : B: Ar
Ion charge : 5: 18
Fraction : 0.95: 0.05

;Irregular grid for code calculation (Z_min:Z_step1:Z_mid:Z_step2:Z_max), (X_min:X_step:X_max), (Y_min:Y_step:Y_max)
code_grid_arr.Z:
0.400, 0.500, 4.400, 0.100, 5.000
code_grid_arr.X:
-0.150, 0.010, 0.150
code_grid_arr.Y:
-0.150, 0.010, 0.150

;Density of electrons, radial position for raw data, m
n_e_raw_r:

* Here should be n_e_raw_r values separated (by one comma and space)

;Density of electrons, raw data, cm-3
n_e_raw:
* Here should be n_e_raw values separated (by one comma and space)

;Density of electrons, errors of the raw data, cm-3
n_e_raw_err:
* Here should be n_e_raw_err values separated (by one comma and space)

;Density of electrons, radial position for smoothed data , m
n_e_r:
* Here should be n_e_r values separated (by one comma and space)

;Density of electrons, smoothed data, cm-3
n_e:
* Here should be n_e values separated (by one comma and space)

;Density of electrons, errors of the smoothed data, cm-3
n_e_err:
* Here should be n_e_err values separated (by one comma and space)

;Temperature of electrons, radial position for raw data, m
t_e_raw_r:
* Here should be t_e_raw_r values separated (by one comma and space)

;Temperature of electrons, raw data, keV
t_e_raw:
* Here should be t_e_raw values separated (by one comma and space)

;Temperature of electrons, errors of the raw data, keV
t_e_raw_err:
* Here should be t_e_raw_err values separated (by one comma and space)

;Temperature of electrons, radial position for smoothed data , m
t_e_r:
* Here should be t_e_r values separated (by one comma and space)

;Temperature of electrons, smoothed data, keV
t_e:
* Here should be t_e values separated (by one comma and space)

;Temperature of electrons, errors of the smoothed data, keV

t_e_err:
* Here should be t_e_err values separated (by one comma and space)

;Effective Charge Z_Eff, radial position for raw data, m
z_eff_raw_r:
0.6783, 0.8978

;Effective Charge. Z_eff, raw data
z_eff_raw:
1.5000, 1.5000

;Effective Charge Z_Eff, errors of the raw data
z_eff_raw_err:
0.0000, 0.0000

;Effective Charge Z_Eff, radial position for smoothed data , m
z_eff_r:
* Here should be z_eff_r values separated (by one comma and space)

;Effective Charge Z_Eff, smoothed data
z_eff:
* Here should be z_eff values separated (by one comma and space)

;Effective Charge Z_Eff, errors of the smoothed data
z_eff_err:
* Here should be z_eff_err values separated (by one comma and space)

*---------- this part of the file is similar for *.abi file and *.abo file----------

End of file:

Appendix D Format of the *.abo output file.

The *.abo file holds all output results as well as all input parameters/data used to run the ALCBEAM simulation. Here the format of this file is discussed.

The example of a DNBI_ALCATOR_1.abo file is shown below. The file structure is preserved. This manual shows additional explanatory comments (beginning with * and highlighted in blue), which are not present in the real file.

;Output data for ALCBEAM code

;File is created on:
Thu Sep 3 10:22:43 2009

;Beam label
beam:
DNBI_ALCATOR

;Shot number
shot:
1070831028

;Time interval t1, t2
time_interv:
1.080, 1.100

;Divergence type
div_type:
RAY TRACING

;Attenuation type
atten_type:
Full attenuation

;Velocity Distribution
vel_dis_type:
NO

;Plasma stopping cross sections source
stop_plsm_cs:
ADAS (stopping)

;Plasma excitation cross sections source
exc_plsm_cs:
ADAS (excitation)

*Here output file contains the all input parameters/data similar to structure shown in appendix C.
The part is separated by the
*---------- this part of the file is similar for *.abi file and *.abo file----------
*---------- this part of the file is similar for *.abi file and *.abo file----------

* next is the fields for output data

;Energy components of the beam, keV
e_beam:
48.802, 24.401, 16.267, 2.711

;Z_grid of the beam, m
z_beam:
* Here should be z_beam values separated (by one comma and space)

;X_grid of the beam, m
x_beam:
* Here should be x_beam values separated (by one comma and space)

;Y_grid of the beam, m
y_beam:
* Here should be y_beam values separated (by one comma and space)

;Calculated Density of the beam, cm-3
n_beam:
* Here should be unformatted data for 4D n_beam array of beam density

;Fraction of the atoms in first excited state, n=2
exc_n2_frac:
* Here should be unformatted data for 4D exc_n2_arr array of fractions of atoms in n=2 (first excited) state

;Fraction of the atoms in second excited state, n=3
exc_n3_frac:
* Here should be unformatted data for 4D exc_n3_arr array of fractions of atoms in n=3 (first excited) state

;X Coordinates of the velocity vector for all beam pores, m
vel_vec_x:
Here should be unformatted data for 3D array (n_x, n_y, n_bml) of X Coordinates of the velocity vector distribution

;Y Coordinates of the velocity vector for all beam pores, m
vel_vec_y:
Here should be unformatted data for 3D array (n_x, n_y, n_bml) of Y Coordinates of the velocity vector distribution

; Polynomial fit coefficients for XY velocity distribution
vel_vec_coef:
Here should be unformatted data for 5D array (n_e, n_z, n_x, n_y, 9) of polynomial fit coefficients for XY velocity distribution.

End of file:

Appendix E Format of the ALCBEAM *.aset settings file.

The LOAD/CONSTRUCT/CALC settings can be saved/loaded into/from a single ascii-text file.

The example of a DNBI_ALCATOR.aset file is shown below. The file structure is preserved.

```
;Run Settings for ALCBEAM code
---------------------------------

;File is created on:
Sun Dec  9 15:16:18 2012
;File is created by:
ALCBEAM (ver. 4.6)

;Load choice
load_choice:
1

;General type
general_type:
1

;General file
general_file:
/home/bespam/alcbeam/DNBI_TEST.abi

;Beam geometry type
beam_geom_type:
1

;Beam geometry file
beam_geom_file:
/home/bespam/alcbeam/DNBI_TEST.abi

;Beam limiters type
beam_lim_type:
1

;Beam limiters file
beam_lim_file:
/home/bespam/alcbeam/DNBI_TEST.abi

;Beam parameters type
beam_param_type:
```

1

;Beam parameters file
beam_param_file:
/home/bespam/alcbeam/DNBI_TEST.abi

;Electron density type
ne_type:
3

;Electron density file
ne_file:
/home/bespam/alcbeam/DNBI_TEST.abi

;Electron temperature type
te_type:
4

;Electron temperature file
te_file:
/home/bespam/alcbeam/DNBI_TEST.abi

;Effective charge type
z_eff_type:
4

;Effective charge file
z_eff_file:
/home/bespam/alcbeam/DNBI_TEST.abi

;Plasma geometry type
plasma_geom_type:
1

;Plasma geometry file
plasma_geom_file:
/home/bespam/alcbeam/DNBI_TEST.abi

;Gas parameters type
gas_type:
1

;Gas parameters file
gas_file:
/home/bespam/alcbeam/DNBI_TEST.abi

;Calculation drig type
grid_type:
1

;Calculation grid file
grid_file:
/home/bespam/alcbeam/DNBI_TEST.abi

;Plasma parameters type
plasma_param_type:
1

;Plasma parameters file
plasma_param_file:
/home/bespam/alcbeam/DNBI_TEST.abi

;Flux surface names
flux_surf_names:
Miller equilib, Hakkarainen eq, skip

;Flux surface array type
flux_surf_arr_type:
0

;Electron density array type
ne_arr_type:
0

;Electron temperature array type
te_arr_type:
0

;Effective charge array type
z_eff_arr_type:
0

;Plasma beam stoping cross section type
stop_plasma_type:
0

;Plasma beam stoppting cross section names
stop_plasma_type_names:
ADAS Delabie (stop), ADAS v3_1 (stop) , ADAS v3_0 (stop) , Suzuki (stop) , skip

;Plasma beam excitation cross section type

exc_plasma_type:
0

;Plasma beam excitation cross section names
exc_plasma_type_names:
ADAS Delabie (exc), ADAS v3_1 (exc) , ADAS v3_0 (exc) , Ian Hutchinson , skip

;Gas density arr type
gas_arr_type:
0

;Beam gas stopping cross section type
stop_gas_type:
0

;Limiters arr type
lim_arr_type:
0

;Used grid aperture names
grid_aper_names:
All apertures, Every 2 aperture , Every 4 aperture , Every 6 aperture , Every 8 aperture , Every 10 aperture , Every 12 aperture , Every 14 aperture , Every 16 aperture , Every 18 aperture , Every 20 aperture , Every 22 aperture

;Used grid aperture type
grid_aper_type:
0

;Divergence model type
div_type:
0

;Divergence model names
div_type_names:
INTERPOLATING, RAY TRACING, ANALYTIC

;Attenuation model type
atten_type:
0

;Attenuation model names
atten_type_names:
Full attenuation , Plasma only, Gas only, Limiters only, Gas + Limiters, Skip attenuation

;Calculate velocity distribution
vel_dis_type:
0

;Calculate velocity distribution names
vel_dis_names:
YES, NO

;Save output type
save_output_type:
2

;Save output filename
save_output_file:
/home/bespam/alcbeam/DNBI_ALCATOR_1.abo

End of file:

Appendix F **Format of the ALCBEAM *.adat data export file**

The *.adat file contains the data and description of a particular graph presented in the graph window.

The example of a DNBI_ALCATOR.adat file is shown below. The file structure is preserved.

```
Exported Data from ALCBEAM plot
---------------------------------

;File is created on: ;
Mon Apr 18 16:32:11 2011
;File is created by:
ALCBEAM (ver. 3.7)

;Plot type
beam line density vs r_major

;Title
Beam Line Density, x10!U10 !Ncm!U-1!N (n=1,2,3: total)

;Xtitle
Major radius, m

;Ytitle
Beam line density, x10!U10 !Ncm!U-1!N

;Text
Beam: DNBI_ALCATOR

;Text
Shot: 1070831028

;Text
Time: [0.408-0.441] sec

;Text
E!Dfull!N: 48.8 keV

;Text
E!Dfull!N: 48.8 keV

;Plotx
  0.661441   0.671213   0.680992   0.690777   0.700568   0.710365
  0.720168   0.729976   0.739789   0.749607   0.759429   0.769257
```

0.779088 0.788925 0.798765 0.808609 0.818456 0.828308
0.838163 0.848021 0.857882 0.867748 0.877615 0.887486
0.897360 0.907236 0.917116 0.926997

;Ploty
0.695294 0.740151 0.788241 0.839026 0.891946 0.949772
1.01090 1.07496 1.14317 1.21564 1.29158 1.37280
1.45686 1.54653 1.63882 1.73664 1.83734 1.94288
2.04876 2.15658 2.26492 2.37180 2.47891 2.57946
2.65029 2.67177 2.68235 2.68988

;Text
E/2: 24.4 keV

;Plotx
0.661441 0.671213 0.680992 0.690777 0.700568 0.710365
0.720168 0.729976 0.739789 0.749607 0.759429 0.769257
0.779088 0.788925 0.798765 0.808609 0.818456 0.828308
0.838163 0.848021 0.857882 0.867748 0.877615 0.887486
0.897360 0.907236 0.917116 0.926997

;Ploty
0.105956 0.116544 0.128249 0.141059 0.154951 0.170485
0.187483 0.205967 0.226260 0.248500 0.272639 0.299172
0.327703 0.358969 0.392399 0.428771 0.467506 0.509133
0.552450 0.597758 0.644530 0.692094 0.740785 0.787585
0.820454 0.829810 0.833028 0.835298

;Text
E/3: 16.3 keV

;Plotx
0.661441 0.671213 0.680992 0.690777 0.700568 0.710365
0.720168 0.729976 0.739789 0.749607 0.759429 0.769257
0.779088 0.788925 0.798765 0.808609 0.818456 0.828308
0.838163 0.848021 0.857882 0.867748 0.877615 0.887486
0.897360 0.907236 0.917116 0.926997

;Ploty
0.297135 0.334035 0.375700 0.422352 0.474186 0.533232
0.599305 0.672837 0.755285 0.847564 0.949995 1.06483
1.19122 1.33242 1.48694 1.65823 1.84458 2.04848
2.26537 2.49639 2.73916 2.99071 3.25211 3.50715
3.68670 3.73622 3.74982 3.75916

;Text

E/18: 2.7 keV

;Plotx
0.661441 0.671213 0.680992 0.690777 0.700568 0.710365
0.720168 0.729976 0.739789 0.749607 0.759429 0.769257
0.779088 0.788925 0.798765 0.808609 0.818456 0.828308
0.838163 0.848021 0.857882 0.867748 0.877615 0.887486
0.897360 0.907236 0.917116 0.926997

;Ploty
0.00875829 0.0112024 0.0143374 0.0183413 0.0234324 0.0299798
0.0383260 0.0489233 0.0624093 0.0795339 0.101158 0.128547
0.162877 0.206118 0.259890 0.326891 0.409209 0.509969
0.630677 0.773887 0.940945 1.13287 1.35147 1.58364
1.75380 1.79913 1.80508 1.80897

Appendix G Format of the ALCBEAM MDSPLUS node of the C-MOD tree

The ALCBEAM output results (along with used input parameters/data) can be saved into the ALCBEAM node of the MDSPLUS database of a particular shot. Here the format of this node is discussed. Beam is DNBI_ALCBEAM by default.

Explanatory comments begin with * and are highlighted in blue.

CMOD.DNB.ALCBEAM.

BESPAM. *This node is always a linux username of the person who run the code

RUN_1. *This node contains the run number described in chapter 3.5.4.

TIME_STAMP *This node contains the time/data when this run was saved

TIME_INTERV *This node contains the [**t1**, **t2**] array. See Appendix A.1.

DIV_TYPE *This node contains the label of type of divergence was used in calculation. See chapter 2.5.3.

ATTEN_TYPE *This node contains the label of type of attenuation was taken into the account. See chapter 2.5.3.

EXC_PLSM_CS *This node contains the label of type of excitation data used in calculation. See chapter 3.5.2.6

STOP_PLSM_CS *This node contains the label of type of plasma stopping cross sections was used in calculation. See chapter 3.5.2.5

INPUT .*A tree contains all input data

CODE_GRID. *This node contains the parameters were used to construct the calculation grid/mesh. See chapter 3.3.2 and Appendix A.2

X_MIN * See chapter 3.3.2and Appendix A.2

X_STEP * See chapter 3.3.2 and Appendix A.2

X_MAX * See chapter 3.3.2 and Appendix A.2

Y_MIN * See chapter 3.3.2 and Appendix A.2

Y_STEP * See chapter 3.3.2 and Appendix A.2

Y_MAX * See chapter 3.3.2 and Appendix A.2

Z_MIN * See chapter 3.3.2 and Appendix A.2

Z_STEP1 * See chapter 3.3.2 and Appendix A.2

Z_MID * See chapter 3.3.2 and Appendix A.2

Z_STEP2 * See chapter 3.3.2 and Appendix A.2

Z_MAX * See chapter 3.3.2 and Appendix A.2

BEAM_GEOM . *This node contains the parameters which describe beam geometry. See chapter 3.3.3 and Appendix A.3

BEAM_APERTUR *This node contains [[x_bml],[y_bml]]. See chapter 3.3.3 and appendix A.3

GRID_AP_DIAM *See chapter 3.3.3 and Appendix A.3

BEAM_PORT *See chapter 3.3.3 and Appendix A.3

GRID_FOCUS *See chapter 3.3.3 and Appendix A.3

MAGNET_SIZE *See chapter 3.3.3 and Appendix A.3

NEUTR_SIZE *See chapter 3.3.3 and Appendix A.3

TANK_CAL *See chapter 3.3.3 and Appendix A.3

TANK_FRONT *See chapter 3.3.3 and Appendix A.3

TANK_MAGNET *See chapter 3.3.3 and Appendix A.3

TANK_SIZE *See chapter 3.3.3 and Appendix A.3

R_GRID *See chapter 3.3.3 and Appendix A.3

Z_GRID *See chapter 3.3.3 and Appendix A.3

PHI_GRID *See chapter 3.3.3 and Appendix A.3

R_WALL *See chapter 3.3.3 and Appendix A.3

Z_WALL *See chapter 3.3.3 and Appendix A.3

PHI_WALL *See chapter 3.3.3 and Appendix A.3

NEUTR_GAS. *This node contains the neutral gas parameters. See chapter 3.3.4 and Appendix A.4

TANK_P *See chapter 3.3.4 and Appendix A.4

TORUS_P *See chapter 3.3.4 and Appendix A.4

PLASMA_GEOM. *This node contains the plasma geometry parameters. See chapter 3.3.5 and Appendix A.5

ELONG *See chapter 3.3.5 and Appendix A.5

R_MAJOR *See chapter 3.3.5 and Appendix A.5

R_MINOR *See chapter 3.3.5 and Appendix A.5

Z_MAJOR *See chapter 3.3.5 and Appendix A.5

TRIANG_L *See chapter 3.3.5 and Appendix A.5

TRIANG_U *See chapter 3.3.5 and Appendix A.5

PLASMA_PARAM. *This node contains the plasma geometry parameters. See chapter 3.3.8 and Appendix A.8

MAIN_ION *See chapter 3.3.8 and Appendix A.8

N_IMPUR *See chapter 3.3.8 and Appendix A.8

IMPUR_TABLE *See chapter 3.3.8 and Appendix A.8

BEAM_PARAM . *This node contains the energy/density beam parameters. See chapter3.3.6 and Appendix A.6

DIV_BML_OPT *See chapter 2.3.6 and Appendix A.6

E_FRAC *See chapter 2.3.6 and Appendix A.6

E_FULL *See chapter 2.3.6 and Appendix A.6

I_BEAM *See chapter 2.3.6 and Appendix A.6

I_FRAC *See chapter 2.3.6 and Appendix A.6

DIV_DISTR_PAR *See chapter 2.3.6 and Appendix A.6

I_DENS_PAR *See chapter 2.3.10 and Appendix A.6

BEAM_LIMITER. *This node contains the beam limiters parameters. See chapter 3.3.7 and Appendix A7

N_LIMITERS *See chapter 3.3.7 and Appendix A.7

LIM_TABLE *See chapter 3.3.7 and Appendix A.7

N_E_PROF . *This node contains the electron density profiles. See chapter 3.3.9 and Appendix A.9

N_E *See chapter 3.3.9 and Appendix A.9

N_E_ER *See chapter 3.3.9 and Appendix A.9

N_E_R *See chapter 3.3.9 and Appendix A.9

N_E_RAW *See chapter 3.3.9 and Appendix A.9

N_E_RAW_ER *See chapter 3.3.9 and Appendix A.9

N_E_RAW_R *See chapter 3.3.9 and Appendix A.9

T_E_PROF. *This node contains the electron temperature profiles. See chapter 3.3.9 and Appendix A9

T_E *See chapter 3.3.9 and Appendix A.9

T_E_ER *See chapter 3.3.9 and Appendix A.9

T_E_R *See chapter 3.3.9 and Appendix A.9

T_E_RAW *See chapter 3.3.9 and Appendix A.9

T_E_RAW_ER *See chapter 3.3.9 and Appendix A.9

T_E_RAW_R *See chapter 3.3.9 and Appendix A.9

Z_EFF_PROF. *This node contains the z_eff profiles. See chapter 3.3.9 and Appendix A.9

Z_EFF *See chapter 3.3.9 and Appendix A.9

Z_EFF_ER *See chapter 3.3.9 and Appendix A.9

Z_EFF_R *See chapter 3.3.9 and Appendix A.9

Z_EFF_RAW *See chapter 3.3.9 and Appendix A.9

Z_EFF_RAW_ER *See chapter 3.3.9 and Appendix A.9

Z_EFF_RAW_R *See chapter 3.3.9 and Appendix A.9

RESULTS. *A tree contains all output data. All the following three 4D arrays are saved as a signal so (**e_beam**, **z_beam**, **x_beam**, **y_beam**) can be extracted from them by DIM_OF ($,$) command

N_BEAM * 4D array of beam density (total for n=1,2,3, all excitation states). See Appendix B.

EXC_N2_FRAC * 4D array of fraction of atoms in n=2 (first excited) state. See Appendix B.

EXC_N3_FRAC * 4D array of fraction of atoms in n=3 (second excited) state. See Appendix B.

Appendix H RAY TRACING simulation

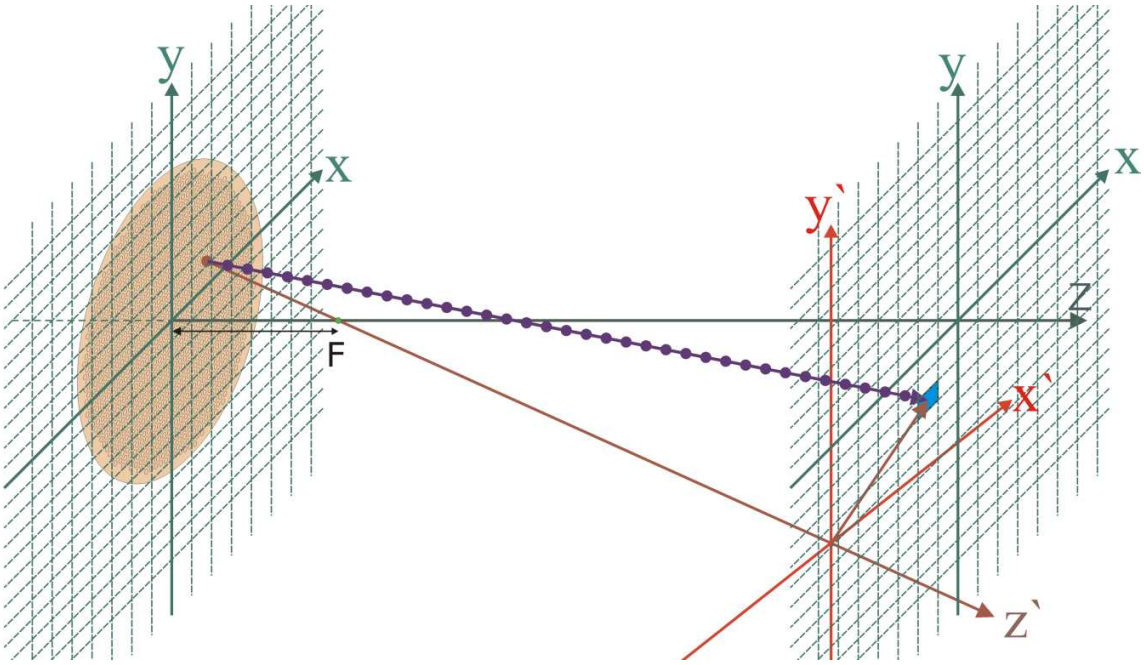
For a detailed description see I.O. Bespamyatnov, W.L. Rowan and K.T. Liao, Comput. Phys. Comm. 183 669-676 (2012)

The divergence model assumes that each pore emits a beamlet of ions along the curvature radius of the external grid. Each beamlet has a density distribution transverse to the trajectory that is Gaussian. The ion current density distribution in cylindrical coordinate system is

$$j(r',\varphi',z')\frac{j_0}{\pi\theta^2 z'^2}\times\exp\left(-\frac{r'^2}{\theta^2 z'^2}\right) \qquad \textit{(3.8.1)}$$

where j_0 is the total current produced by single grid, θ is the half divergence angle of a beamlet.

A pencil simulation code was developed to calculate the current density distribution at any distance from grid. The code launches a beamlet from each of the grid pore. The beamlet follows a straight line trajectory perpendicular to the grid at the location from which it is launched. The beamlet current transverse to the beamlet trajectory is a Gaussian (see eqn. (3.8.1)). The total beam current profile is taken as a sum of the contributions from all apertures.



$$\frac{dn_{beam}}{n_{beam}} = -(n_{gas} \cdot \sigma_{stop-gas} + n_e \cdot \sigma_{stop-plasma}) \cdot dl + C_{\mathrm{lim}iter}$$

Appendix I INTERPOLATION simulation

For a detailed description see I.O. Bespamyatnov, W.L. Rowan and K.T. Liao, Comput. Phys. Comm. 183 669-676 (2012)

Appendix J Analytical model for beam divergence.

This mode can only be used for circular beams with equal x and y focal distances and equal x and y beamlet divergences. The geometric model of the ion emission from the grid is shown in Figure 3.8.1. The beam centerline is directed along the *Z*-axis, which is the grid axis of symmetry. The beam can be described using cylindrical coordinates (r, φ, z) with the r-φ plane transverse to the beam trajectory and the origin coincided to center of the grid. The curvature radius of the grid is *R*, which is much larger than radial dimension of the grid *a*

$$R >> a \qquad (3.8.1)$$

This inequality leads to an important simplification. The grid is represented as a circular disk of radius *a*.

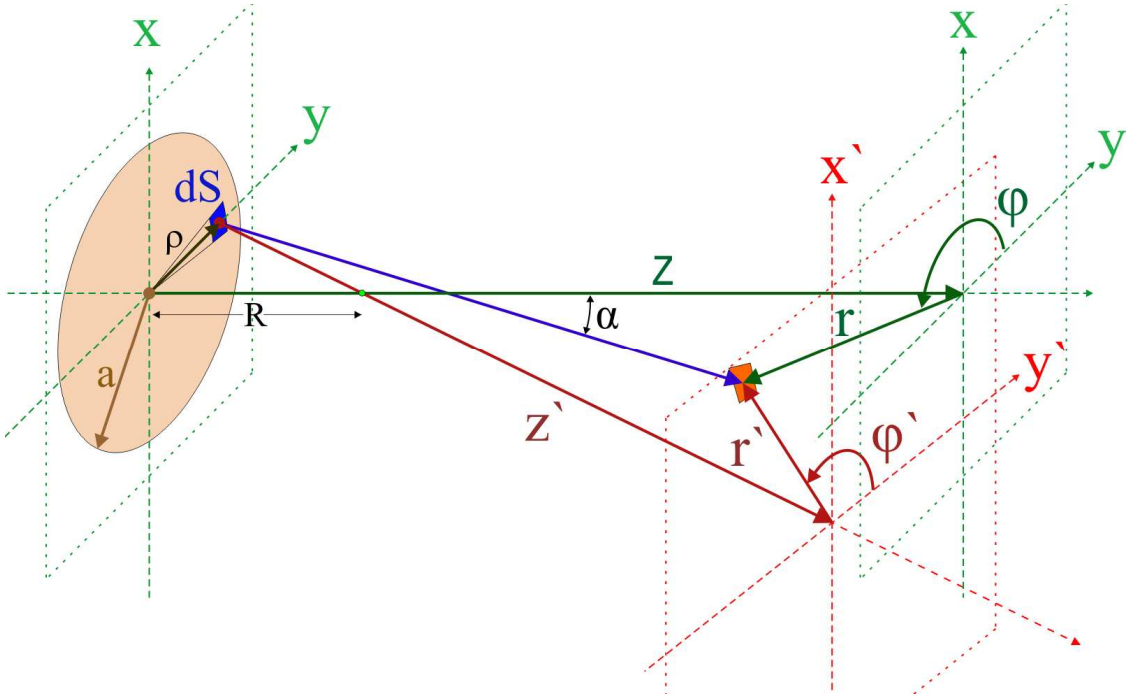


Figure 3.8.1: The geometrical model of the ions emission from the grid. The beam centerline is along Z-axis.

Due to the large number of pores, the beam density at any spatial point (r, φ, z) can be represented as a sum of individual contributions from each elemental part of the focusing grid surface. It is assumed that the grid curvature, which determines the beam focal length R, is uniform over the grid surface. Each pore is represented by an elemental surface dS, which emits a beamlet of ions in direction Z'. The elemental current contribution from the surface dS at point (r', φ', z') is calculated according to (3.8.1).

$$dj = \frac{dj_0}{\pi\theta^2 z'^2} \times \exp\left(-\frac{r'^2}{\theta^2 z'^2}\right) = \frac{J_0}{2\pi a^2} \cdot \frac{dS}{\pi\theta^2 z'^2} \times \exp\left(-\frac{r'^2}{\theta^2 z'^2}\right) \qquad \textbf{(3.8.2)}$$

where a is the grid radius and J_0 is the total current of the grid.

To find the current density in the (r, φ, z) coordinate system, an appropriate $(r', \varphi', z') \rightarrow (r, \varphi, z)$ transformation must be found. In Cartesian coordinates, this transformation $(x', y', z') \rightarrow (x, y, z)$ is represented by one translation and one rotation

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \hat{A}\left[\begin{pmatrix} x \\ y \\ z \end{pmatrix} + \bar{b}\right], \quad \hat{A} = \begin{pmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{pmatrix}, \quad \bar{b} = \begin{pmatrix} \rho \\ 0 \\ 0 \end{pmatrix} \qquad \textbf{(3.8.3)}$$

where $\hat{A}$ is the matrix of rotation around *y* axis by angle α, $\bar{b}$ is the translation vector along *x* by distance ρ.

Combining both transformations gives:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} (x-\rho)\cos\alpha + z\sin\alpha \\ y \\ z\cos\alpha - (x-\rho)\sin\alpha \end{pmatrix} \qquad \textbf{(3.8.4)}$$

The transformation from Cartesian to cylindrical coordinates follows

$$\begin{pmatrix} r' \\ \varphi' \\ z' \end{pmatrix} = \begin{pmatrix} \sqrt{x'^2 + y'^2} \\ \tan^{-1}\left(\dfrac{x'}{y'}\right) \\ z' \end{pmatrix} \qquad \textbf{(3.8.5)}$$

Returning to eqn. (3.8.2), we only need to find the transformation for $\frac{r'^2}{z'^2}$ (exponential coefficient) and z'^2 (denominator). Using the fact that

$$\cos^2\alpha = \frac{R^2}{R^2+\rho^2} \quad \textit{and} \quad \tan^2\alpha = \frac{\rho^2}{R^2}, \qquad \textbf{(3.8.6)}$$

and applying some algebraic manipulation, the needed transformations are found to be

$$\frac{r'^2}{z'^2} = \frac{\frac{r^2}{z^2} + \rho^2\left(\frac{1}{z} - \frac{1}{R}\right)^2 - \frac{2r\rho\sin\varphi}{z}\left(\frac{1}{z} - \frac{1}{R}\right) + \frac{r}{z^2}^2\frac{\rho^2}{R^2}\cos^2\varphi}{1 + \frac{r^2}{z^2}\frac{\rho^2}{R^2}\sin^2\varphi - \frac{\rho^2}{z^2}\frac{\rho^2}{R^2} - 2\frac{r\rho}{zR}\sin\varphi + \frac{2\rho^2}{zR}} \qquad \textit{(3.8.7)}$$

$$z'^2 = z^2\left(1 + \frac{r^2}{z^2}\frac{\rho^2}{R^2}\sin^2\varphi - \frac{\rho^2}{z^2}\frac{\rho^2}{R^2} - 2\frac{r\rho}{zR}\sin\varphi + \frac{2\rho^2}{zR}\right)\times\frac{R^2}{R^2+\rho^2}$$

At this step, it is necessary to introduce $r, \rho << R, z$, which follows from the inequality (3.8.1) and means that we are looking for solutions far away from the grid. Making the substitution: $\frac{1}{z*} = \frac{1}{z} - \frac{1}{R}$, the eqns (3.8.7) break up into several terms of different orders of magnitude.

$$\frac{r'^2}{z'^2} = \frac{r^2}{z^2} + \frac{\rho^2}{z*^2} - \frac{2r\rho\sin\varphi}{z\,z*} + O\left(\frac{\rho^2 r^2}{z^2R^2}\right) + O\left(\frac{\rho r^3}{z^3R}\right) + O\left(\frac{\rho^3 r}{z^3R}\right) + O\left(\frac{\rho^2 r^2}{z^3R}\right)$$

$$z'^2 = z^2\left(1 + O\left(\frac{\rho r}{Rz}\right) + O\left(\frac{\rho^2}{zR}\right) + O\left(\frac{\rho^2 r^2}{z^2R^2}\right) + O\left(\frac{\rho^4}{z^2R^2}\right)\right) \qquad \textit{(3.8.8)}$$

Dropping out the small terms, the eqn. (3.8.2) transforms to

$$dj(r,\varphi,z) \approx \frac{J_0}{2\pi a^2}\cdot\frac{dS}{\pi\theta^2 z^2}\cdot\exp\left(-\frac{r^2}{\theta_0{}^2 z^2}\right)\cdot\exp\left(-\frac{\rho^2}{\theta_0{}^2 z*^2}\right)\cdot\exp\left(\frac{2r\rho\sin\varphi}{\theta_0{}^2 z\,z*}\right) \qquad \textit{(3.8.9)}$$

The total current density at a point (r, φ, z) is the sum of the contributions from all pores, which is mathematically represented as an integral over the grid surface.

$$j(r,z)=\int_{dS} dj(r,\varphi,z)\approx$$

$$\approx\int_0^a\int_0^{2\pi}\frac{J_0\,d\varphi\,\rho\,d\rho}{\pi^2 a^2\,{\theta_0}^2 z^2}\cdot\exp\left(-\frac{r^2}{{\theta_0}^2 z^2}\right)\cdot\exp\left(-\frac{\rho^2}{{\theta_0}^2 z^{*2}}\right)\cdot\exp\left(\frac{2r\rho\sin\varphi}{{\theta_0}^2 z\,z^*}\right)= \qquad \textbf{(3.8.10)}$$

$$=\frac{2J}{\pi a^2\,{\theta_0}^2 z^2}\cdot\exp\left(-\frac{r^2}{{\theta_0}^2 z^2}\right)\times\int_0^a\cdot\exp\left(-\frac{\rho^2}{{\theta_0}^2 z^{*2}}\right)\cdot I_0\left(\frac{2r\rho}{z\,z^*\theta^2}\right)\rho\,d\rho$$

where $I_0(x)=\frac{1}{2\pi}\int_0^{2\pi}\exp(x\sin\varphi)\,d\varphi$ is the modified Bessel function of the first kind and $\frac{1}{z^*}=\frac{1}{z}-\frac{1}{R}$.